

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

# **Modul pro ovládaní výstupů chytrým telefonem připojeným do systému iDUM.**

## **Modul for Handling Outputs by Smart Telephone Connecting to System iDUM**

# Zadání diplomové práce

Student: **Bc. Martin Kriváček**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Modul pro ovládání výstupů chytrým telefonem připojeným do systému iDUM.  
Modul for Handling Outputs by Smart Telephone Connecting to System iDUM.

Jazyk vypracování: čeština

## Zásady pro vypracování:

1. Specifikace požadavků kladených na modul :
  - 1.1 Snadné ovládání zařízení typu vjezdová brána, osvětlení apod. pomocí WIFI, s automatickým spouštěním aplikace v systému Android.
  - 1.2 Modul musí být schopen jak samostatné funkce, tak i funkce v systému iDUM.
  - 1.3 Modul má schopnost detekovat přiřazené zařízení (na základě MAC adresy WIFI), zareaguje vysláním zprávy do systému iDUM s MAC adresou zařízení.
  - 1.4 Pokud je modul detekován mobilním telefonem s OS Android, tak se v mobilním telefonu spustí ovládací aplikace uživatelského rozhraní. Tato aplikace bude schopna sepnout výstupy modulu (např. relé) a odeslat předdefinovanou zprávu po lince iDum.
2. Na základě specifikovaných požadavků navrhnete řešení modulu po stránce technické i programové.
3. Vyberte vhodné Wi-Fi rozhraní a připojte ho k modulu.
4. Vytvořte aplikaci pro mobilní telefon pod OS Android s vhodným uživatelským rozhraním.
5. Předved'te provozuschopnost realizovaného funkčního vzoru.
6. Vytvořte úplnou technickou dokumentaci modulu.

## Seznam doporučené odborné literatury:

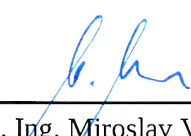
- [1] Pohorský, J. :HDO - hromadné dálkové ovládání. BEN, 2011, ISBN 978-80-7300-347-0.
- [2] Jurick, D. : Velká kniha tipů a triků pro iPhone. Computer Press, 2009, ISBN 978-80-251-2645-5.
- [3] Geisler, M. : Bezdrátové ovládání spotřebičů. BEN, ISBN 80-7300-013-X.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radek Novák, Ph.D.**


Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



---

doc. Ing. Miroslav Vozňák, Ph.D.  
vedoucí katedry



---

prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne zdroje a publikácie, z ktorých som čerpal.

V Ostrave 20. apríla 2016

..........



Rád by som poďakoval Ing. Radkovi Novákovi, Ph.D. a Ing. Michalovi Jahelkovi, Ph.D. za odbornú pomoc pri riešení diplomovej práce.

## **Abstrakt**

Diplomová práca je zameraná na návrh a realizáciu modulu pripojeného do systému iDŕm, kde výstupy tohto modulu sú ovládané pomocou inteligentného telefónu.

V práci sú teoreticky popísané jednotlivé prvky modulu a platformy použité pri vývoji, ako aj samotný systém iDŕm.

Praktická časť práce obsahuje návrh modulu, logické a fyzické zapojenie. Obsiahnutý je aj popis programovej časti pre mikroprocesor PIC a Wi-Fi modul ESP8266. Pre ovládanie tohto modulu bola vytvorená aplikácia pre Android OS.

V závere práce sú popísané problémy, ktoré sa objavili pri vývoji a ďalší možný vývoj tohto modulu.

**Kľúčové slová:** PIC, ESP8266, inteligentný dom, iDŕm, Android, Wi-Fi

## **Abstract**

Master thesis is focused to design and implementation of module connected to system iDŕm, where outputs of this module are controlled by smart phone application.

This thesis contains theoretical description of parts used in this module and platforms used during development, as well as the system iDŕm itself.

Practical part of this thesis contains module design, logical and physical connections. Included is also a description of the program for the PIC microcontroller and Wi-Fi module ESP8266. Application for Android OS was created to control this module.

In end of this thesis are described problems encountered in the development and possible future development of this module .

**Key Words:** PIC, ESP8266, Home automation, iDŕm, Android, Wi-Fi

# Obsah

<b>Zoznam použitých skratiek a symbolov</b>	<b>9</b>
<b>Zoznam obrázkov</b>	<b>11</b>
<b>Zoznam tabuliek</b>	<b>13</b>
<b>1 Úvod</b>	<b>14</b>
<b>2 Inteligentný dom</b>	<b>15</b>
2.1 Popis inteligentných domov . . . . .	15
2.2 iDům . . . . .	15
<b>3 Wi-Fi modul ESP8266</b>	<b>19</b>
3.1 Wi-Fi SoC ESP8266 . . . . .	19
3.2 ESP8266 DevKit . . . . .	20
3.3 Možnosti programovania . . . . .	21
<b>4 Jednočipové mikrokontroléry PIC</b>	<b>24</b>
4.1 Rada mikrokontrolérov PIC18 . . . . .	24
4.2 Vývojové prostredie MPLAB X . . . . .	25
4.3 Programátor PICKit 3 . . . . .	26
<b>5 OS android</b>	<b>27</b>
5.1 Architektúra . . . . .	27
5.2 Vývoj aplikácií . . . . .	28
5.3 Android Studio . . . . .	29
<b>6 Realizácia modulu</b>	<b>31</b>
6.1 Návrh modulu . . . . .	31
6.2 Logické zapojenie . . . . .	32
6.3 Fyzické zapojenie . . . . .	34
6.4 Program mikrokontroléra PIC . . . . .	37
6.5 Program Wi-Fi modulu ESP8266 . . . . .	42
6.6 Aplikácia pre OS Android . . . . .	42
<b>7 Popis funkčnosti modulu</b>	<b>50</b>
7.1 Spustenie . . . . .	50
7.2 Ovládanie modulu . . . . .	51
7.3 Ukončenie . . . . .	53
7.4 Zabezpečenie . . . . .	53

7.5	Android 5+ a prístupové body bez internetového pripojenia . . . . .	54
7.6	Vylepšenie modulu . . . . .	55
<b>8</b>	<b>Záver</b>	<b>56</b>
	<b>Literatúra</b>	<b>57</b>
	<b>Prílohy</b>	<b>59</b>
<b>A</b>	<b>Schéma a výpis elektronických súčiastok</b>	<b>59</b>
<b>B</b>	<b>Elektronická príloha</b>	<b>61</b>

## Zoznam skratiek a symbolov

AES	– Advanced Encryption Standard
A/D	– Analog / Digital
ADT	– Android Development Tool
AT	– Attention
CAN	– Controller Area Network
CMOS	– Complementary Metal–Oxide–Semiconductor
DHCP	– Dynamic Host Configuration Protocol
DIL	– Dual In-line Package
EEPROM	– Electrically Erasable Programmable Read-Only Memory
FTP	– File Transfer Protocol
GND	– Ground
GPIO	– General-Purpose Input/Output
GUI	– Graphical User Interface
HTTP	– HyperText Transfer Protocol
I <sup>2</sup> C	– Inter-Integrated Circuit
I <sup>2</sup> S	– Inter-IC Sound
ID	– Identifikation
IDE	– Integrated Development Environment
IoT	– Internet of Things
IP	– Internet Protocol
IPv4	– Internet Protocol version 4
LCD	– Liquid-Crystal Display
LIN	– Local Interconnect Network
MIPS	– Million Instructions Per Second
PIC	– Programmable Interface Controllers
PWM	– Pulse-Width Modulation
SDIO	– Secure Digital Input Output
SDK	– Software Development Kit
SoC	– System on a Chip
SoftAP	– Software Access Point
SPI	– Serial Peripheral Interface
SSID	– Service Set Identifier
TCP	– Transmission Control Protocol
TKIP	– Temporal Key Integrity Protocol
UART	– Universal Asynchronous Receiver/Transmitter
UDP	– User Datagram Protocol
USART	– Universal Synchronous/Asynchronous Receiver/Transmitter

USB	– Universal Serial Bus
WEP	– Wired Equivalent Privacy
WPA	– Wi-Fi Protected Access
WPA2	– Wi-Fi Protected Access II

## Zoznam obrázkov

1	Možné zapojenie systému iDům . . . . .	15
2	Model zbernice systému iDům . . . . .	16
3	Komunikačné rozhranie . . . . .	16
4	Dokončený prenos v systéme iDům . . . . .	17
5	Priebeh budenia . . . . .	17
6	Proti-kolízna ochrana . . . . .	18
7	ESP-01 . . . . .	19
8	ESP-07 . . . . .	19
9	ESP-12-F . . . . .	19
10	Schéma modulu ESP8266 . . . . .	20
11	NodeMCU DevKit ESP8266 . . . . .	20
12	Nastavenia Arduino IDE . . . . .	22
13	Board Manager . . . . .	22
14	Nastavenia IDE pre modul . . . . .	23
15	Hardvardská architektúra . . . . .	24
16	Mikroprocesory PIC18 . . . . .	25
17	Programátor PICKit 3 . . . . .	26
18	Architektúra Andorid OS . . . . .	27
19	Android Studio IDE . . . . .	29
20	Logické zapojenie modulu . . . . .	32
21	Popis komunikácie medzi jednotlivými prvkami . . . . .	33
22	Schéma zapojenia modulu . . . . .	34
23	Realizácia modulu na nepájivom poli . . . . .	34
24	Konektor pre programovanie mikrokontroléra . . . . .	35
25	Konektor pre programovanie mikrokontroléra . . . . .	35
26	Spätná väzba reprezentovaná tlačidlami . . . . .	36
27	Výstupy reprezentované LED diódami . . . . .	36
28	Prispôsobenie sériovej komunikácie pre jednovodičovú zbernicu . . . . .	37
29	iDům Konfigurátor . . . . .	37
30	AppCompat menu . . . . .	43
31	Konfiguračná obrazovka . . . . .	43
32	Notifikácia oznamuje, že telefón je pripojený k modulu . . . . .	46
33	Notifikácia oznamuje vyhľadávanie modulu . . . . .	46
34	GUI obslužnej časti aplikácie . . . . .	47
35	GUI obslužnej časti aplikácie . . . . .	47
36	Zobrazená správa zo systému iDům . . . . .	49
37	Nastavenia aplikácie . . . . .	51

38	Pripojenie k modulu . . . . .	51
39	Prenos správ bez systému iDŕm . . . . .	52
40	Priebeh otvorenia brány so systémom iDŕm . . . . .	53
41	Vzdialenie sa z dosahu modulu . . . . .	54
42	Schéma zapojenia modulu . . . . .	59



## Zoznam tabuliek

2	Definícia spätnej väzby . . . . .	48
3	Tabuľka konfigurácie EEPROM pamäte modulu . . . . .	50
4	Správy medzi modulom a systémom iDům . . . . .	53

# 1 Úvod

Cieľom diplomovej práce bolo vytvorenie modulu pre ovládanie výstupov inteligentným telefónom. Modul by mal byť schopný funkcie buď samostatne alebo v kombinácii so systémom iDŕm.

Spôsob využitia tohto modulu môže byť napríklad na otváranie vstupnej brány na pozemok bez nutnosti vystúpiť z auta.

Navrhnutý modul vysiela prístupový bod, mobilná aplikácia dokáže tento prístupový bod automaticky rozpoznať, keď sa nachádza v jeho okolí a vyvolať obslužnú časť, ktorá sa zobrazí na displeji mobilného telefónu. V tejto časti môžeme ovládať výstupy modulu stlačením tlačidla. Spätná väzba modulu je zobrazovaná na displeji informujúca užívateľa o aktuálnom stave.

V prípade, že je modul pripojený k systému iDŕm, môže byť užívateľ aplikácie informovaný o rôznych udalostiach formou textových správ, ktoré prídu z tohto systému a sú zobrazené v aplikácii. Taktiež sú do systému iDŕm posielané MAC adresy klientov z dôvodu bezpečnosti pri každom pripojení a odpojení.

V prvých kapitolách práce sú postupne spracované teoretické informácie o systéme iDŕm, použitých vývojových nástrojoch a samostatných častiach modulu. Od kapitoly 6 je popísaný výber prvkov obvodu a realizácia fyzického zapojenia tak aj vývoj programovej časti pre mikrokontrolér PIC a mobilnú aplikáciu.

V kapitole 7 je popis funkčnosti modulu s príkladmi komunikácie a možnosťami ovládania modulu. V závere tejto kapitoly sú zhrnuté problémy, ktoré sa vyskytli a smer ďalšieho vývoju modulu.

## 2 Inteligentný dom

### 2.1 Popis inteligentných domov

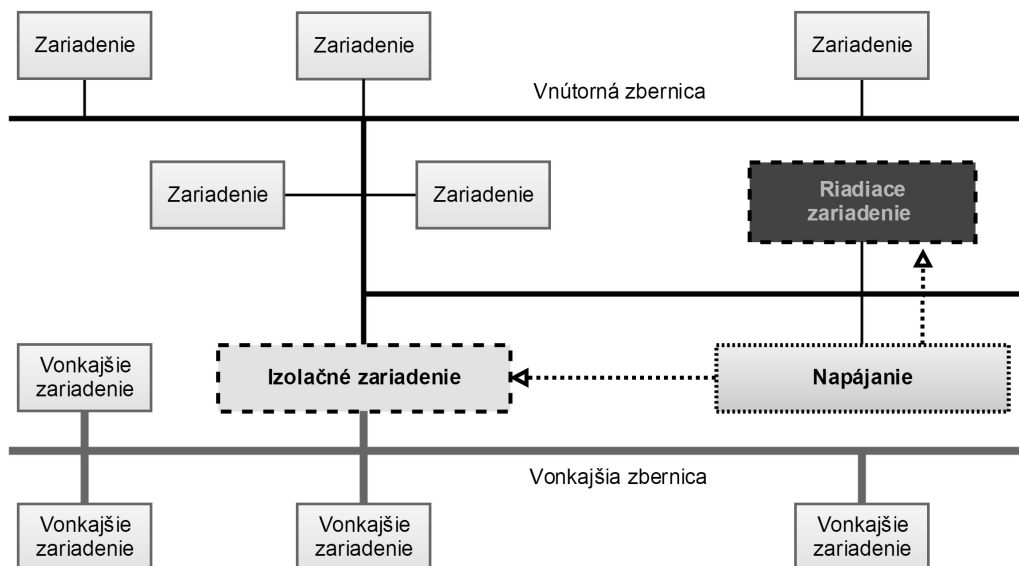
Inteligentné domy sú v súčasnej dobe čoraz častejším javom. Je to hlavne v dôsledku snahy integrovať zariadenia každodenného života do jedného centrálného systému a to či už z dôvodu bezpečnosti, šetrenia energie, efektívnosti alebo zvýšenia pohodlia.

V inteligentnom dome je možné ovládať jedným systémom osvetlenie, vykurovanie či chladenie, tieniacu techniku, brány, okná a zavlažovanie. Zároveň dokáže zastrešiť rôzne technológie, tak aby ich prevádzka bola efektívna a navzájom spolupracovali.

### 2.2 iDům

Systém iDům je systém inteligentného domu pracujúceho na štvorvodičovej zbernici.

Zbernica môže mať ľubovoľnú štruktúru bez nutnosti ukončenia zbernice. Každé jednotlivé zariadenie má svoje jedinečné identifikačné číslo. V systéme môže byť teoreticky celkom 255 rôznych zariadení, ale pre špeciálne použitie v budúcnosti je toto číslo obmedzené na 250. Systém používa 4 vodiče, dva napájacie, jeden pre signalizáciu a jeden pre prenos špecifickej informácie (analogový signál - zvuk/video). To umožňuje použitie jednoduchých a lacných prvkov, ktoré sa nezaujímajú o zložitý prenos špecifickej informácie a používajú len pomalý signalizačný vodič (vypínače, senzory, relé a pod.).



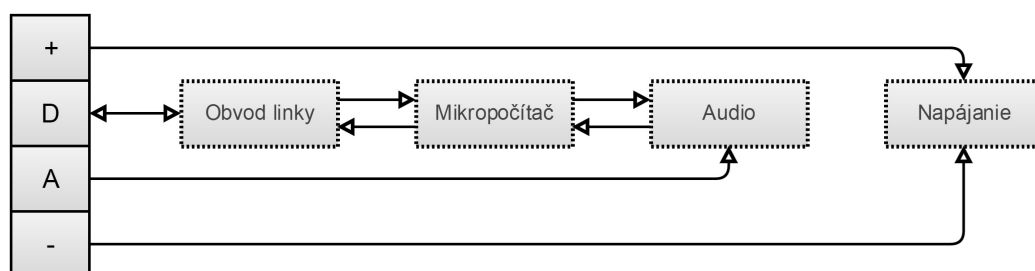
Obr. 1: Možné zapojenie systému iDům

Štandardný prenos dát nieje kódovaný a aby nemohol nikto iný ovládať alebo vyradiť systém z vonkajšieho prostredia nachádza sa v obvode filtračné/izolačné zariadenie, ktoré púšťa len

povolené informácie a naviac opticky oddeluje zbernicu, takže skrat alebo pripojenie vysokého napätia nevyradí alebo nezničí vnútorný systém. [1]

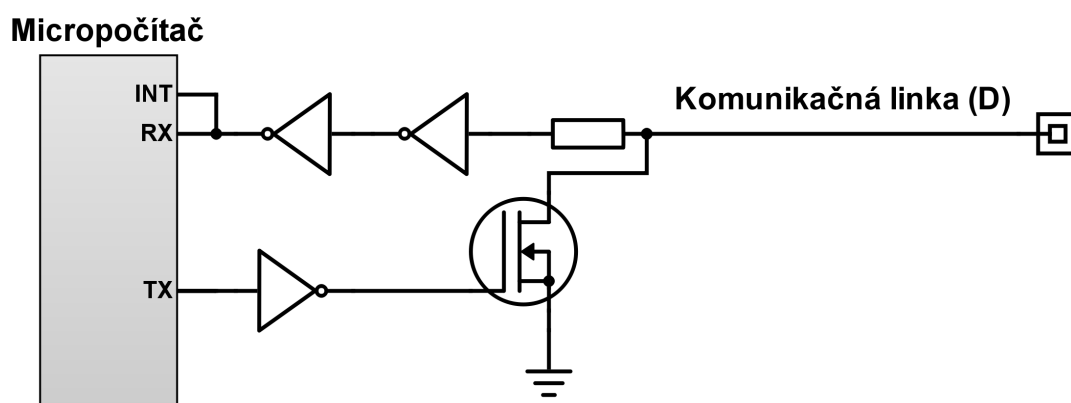
### 2.2.1 Spoločná zbernica iDŕm

Systém inteligentného domu je postavený na štvorvodičovej zbernici. Všetky prvky systému sú prepojené pomocou tejto zbernice. Porucha jedného prvku môže spôsobiť nefunkčnosť celého systému, nie však poškodenie zbernice alebo ostatných prvkov systému. Prerušením zbernice dôjde k odpojeniu všetkých prvkov za prerušením. [1]



Obr. 2: Model zbernice systému iDŕm

K napájaniu zariadení pomocou zbernice je použité napätie 15V. Dátový komunikačný vodič pracuje s úrovňami signálu CMOS 5V. Vstupný signál je tvarovaný Schmittovým obvodom. Vstupné hradlo je chránené 10k $\Omega$  rezistorom proti chybe pri montáži alebo skratu na zbernici. Pokojový stav na zbernici je reprezentovaný vysokou úrovňou. Nízka úroveň je dosiahnutá zopnutím výkonového tranzistoru MOSFET, ktorý skratuje zbernicu do nuly. Upravený vstup zbernice je pripojený na sériový vstup RX a prerušovací vstup INT, kvôli možnosti prebúdzat mikropočítač z režimu spánku. [1]

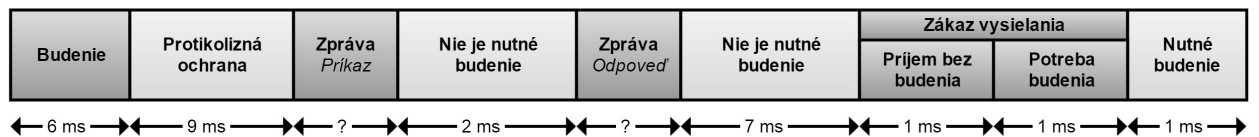


Obr. 3: Komunikačné rozhranie

### 2.2.2 Komunikačný protokol

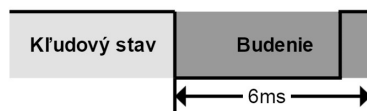
Požiadavky na komunikačný protokol:

- Prenos dát zabezpečený voči strate
- Podpora prenosu správ o rôznej dĺžke
- Detekcia kolízií
- Vyvarovanie sa kolíziám na začiatku prenosu
- Vylúčenie vplyvu dĺžky zbernice
- Prebudenie zariadenia z režimu spánku



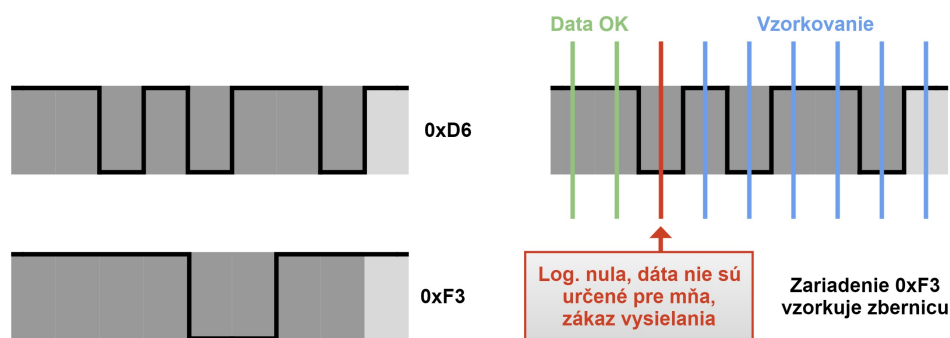
Obr. 4: Dokončený prenos v systéme iDum

Aby bola zaistená nízka spotreba, všetky zariadenia sú v režime spánku, pokiaľ sa neprenášajú žiadne dáta signalizačným vodičom. Ako prvá musí byť prevedená sekvencia "**budenie**". Budiaci signál je  $5\text{ ms}$  v logickej nule a  $1\text{ ms}$  v logickej jednotke. Všetky zariadenia sa musia prebudiť do  $5\text{ ms}$  od prijatia zostupnej hrany signálu. [1]



Obr. 5: Priebeh budenia

Proti-kolízna ochrana zaistuje vysielanie len jedného zariadenia. Každé zariadenie, ktoré potrebuje vysielat na zbernici začne vysielat svoje identifikačné číslo od najvyššieho bitu k najnižšiemu. Dĺžka jedného bitu je  $1\text{ ms}$ . Pokiaľ je vysielaná logická jednotka ale zariadenie rozpozná logickú nulu, tak sa vysielania ďalej neúčastní. Zariadenia s nižším číslom majú prednosť pred zariadeniami s vyšším identifikačným číslom. Z toho dôvodu má riadiaca jednotka systému identifikačné číslo 1, zabezpečovacie prvky nízke čísla a nedôležité zariadenia ako senzory teploty vysoké čísla. Posledný bit proti-kolíznej ochrany je vždy logickej jednotke pre prípravu na príjem správy. [1]



Obr. 6: Proti-kolízna ochrana

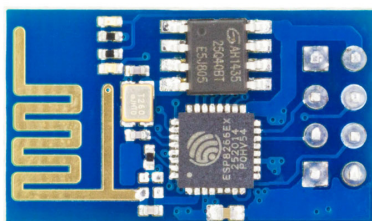
Následne sa vysielajú "správy". Dĺžka správy môže byť obecné rôzna. Správa sa vysielá štandardným komunikačným protokolom používaným na sériovej linke s pokojovou úrovňou v logickej jednotke a nízkou rýchlosťou *19 200 bit/s*. Vysielanie začína štart bitom (log. 0), prenáša sa 8 bitov a končí stop bitom (log. 1). Vďaka nízkej prenosovej rýchlosti nie je nutná špeciálna ochrana pred odrazmi na vedení.

Mikrokontroléry môžu byť pomalé a pri vysielaní správy môže prísť vplyvom prerušenia ku krátkodobej prestávke vo vysielaní. Táto prestávka však nesmie byť dlhšia ako *1,5 ms* od posledného prijatého znaku (stop bitu). Pokiaľ je prekročený čas *2 ms* od posledného prijatého bitu, je **vysielanie považované za ukončené** a ostatné zariadenia môžu pristupovať k zbernici. Vysielanie je taktiež považované za ukončené pokiaľ je prijatá celá správa. Väčšinou nasleduje **odpoveď zariadenia**. Táto odpoveď musí byť prenesená v dobe do *4 ms*. V opačnom prípade vysielacie zariadenie zopakuje správu. V prípade, že v čase *1 ms* po dokončení správy pristúpi k zbernici viacero zariadení, môže dôjsť ku kolíziám. Každé zariadenie však pred samotným vysielaním čaká čas približne  $t = 4\mu s \cdot (DevID - 1)$ . Po uplynutí tohto času monitorujú zbernicu a ak už vysielanie prebieha k zbernici nepristúpi. Pokiaľ zariadenie pristúpi v nasledujúcom časovom okamihu (po uplynutí *1 ms* a viac), toto obmedzenie neplatí. Nenachádza sa tu štandardne dlhá proti-kolízna ochrana, aby sa zrýchlila výmena dát (proti-kolízna ochrana by systém spomalila vždy o *9ms*). V prípade kolízie sa všetky zariadenia odpoja od zbernice a počkajú *6 ms*. Potom začnú vysielajú znovu budiaci impulz a proti-kolíznu ochranu. Pokiaľ zariadenie odpovie, môže odpovedať do *5 ms* od dokončenia posledného prenosu, **bez nutnosti vykonať budúcu sekvenciu**.

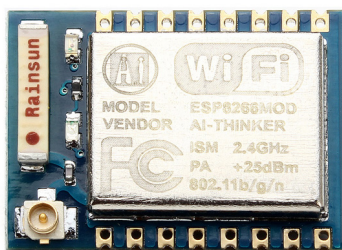
Pokiaľ zariadenie neodpovie, nastáva **zákaz štartu vysielania** či už znaku alebo budúcej sekvencie. Zariadenie je nastavené po dobu *1 ms* pre príjem dát bez budenia (zariadenie môže začať vysielajú tesne pred uplynutím *5 ms*, ale dáta prídu až po zákaze štartu). Pokiaľ by nebol zákaz štartu vysielania, došlo by k problémom. Po uplynutí tejto *1 ms* sú zariadenia nastavená na príjem s budením, ale vysielanie je taktiež zakázané. Za ďalšiu *1 ms* je už štart vysielania povolený, ale vysielajú sa budiaca sekvencia aj proti-kolízna ochrana. Pokiaľ nedôjde k vysielaniu prejdú zariadenia do režimu spánku. [1]

### 3 Wi-Fi modul ESP8266

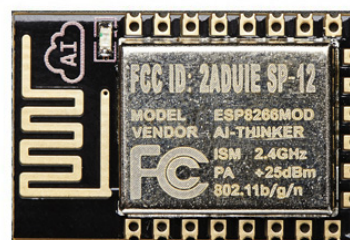
Modul ESP8266 je populárny, lacný Wi-Fi čip, ktorý poskytuje kompletný TCP/IP zásobník, ako aj schopnosti mikrokontroléra produkované Šanghajskou firmou Espressif Systems. Tento čip bol odpoveďou na stále sa rozvíjajúcu oblasť IoT. Po prvý krát sa dostal do povedomia v Auguste 2014, kedy externý výrobca AI-Thinker uviedol na trh modul ESP-01. Modul bol z počiatku označovaný za Serial-to-Wi-Fi adaptér a v základnom stave prichádza s jednoduchým rozhraním používajúcim AT príkazy.



Obr. 7: ESP-01



Obr. 8: ESP-07



Obr. 9: ESP-12-F

Koncom Októbra 2014 spoločnosť Espressif uvoľnila SDK, ktoré umožňovalo programovať priamo čip a používať tento modul samostatne bez pridania ďalšieho mikrokontroléra. Na trh bolo uvedených celkovo 15 rôznych verzií, ktoré sa vzájomne líšia množstvom vyvedených pinov, veľkosťou flash pamäte, tienením a typom antény.

#### 3.1 Wi-Fi SoC ESP8266

ESP8266 je postavený na 32-bitovom RISC procesore Tensilica Xtensa LX106 bežiacom na 80 MHz. Základné parametre:

- Wi-Fi:

Protokoly: 802.11b/g/n

Frekvenčný rozsah: 2.400 - 2.4835 GHz

- Hardware:

Periférie: UART / SDIO / SPI / I<sup>2</sup>C / I<sup>2</sup>S / GPIO / PWM

Operačné napätie: 3.0 - 3.6 V

Operačný prúd: Priemerne 80 mA

- Software

Wi-Fi mód: klient / softAP / softAP + klient





Následné programovanie je veľmi podobné ako pri platforme Arduino. Stačí pripojiť modul pomocou USB kábla k počítaču. Po nainštalovaní ovládača počítač vyhodnotí toto zariadenie ako sériový port a následne môžeme do flash pamäte nahráť požadovaný program. [3]

### 3.3 Možnosti programovania

Zo základným firmvérom poskytnutým od výrobcu čipu spoločnosti Espressif môžeme komunikovať z modulom pomocou AT príkazov podobne ako s telefónnym modemom. K dispozícii je pomerne veľká škála príkazov a jedná sa o najjednoduchší spôsob, akým môže komunikovať mikrokontrolér s týmto modulom.

Pokiaľ však chceme od modulu špecifické správanie, môžeme nahradiť originálny firmvér naším vlastným alebo nahráť firmvér, ktorý vytvorila tretia strana.

#### 3.3.1 NodeMCU firmvér

Jedná sa o firmvér vytvorený súčasne s NodeMCU DevKit-om. Po nahradení originálneho firmvéru sa funkčnosť a správanie mení len zmenou skriptov uložených na flash pamäti. Skriptovací jazyk je založený na Lua 5.1.4. Týmto spôsobom sa jednoducho dá využiť potenciál tohto modulu a môžeme ho požívať samostatne.

Firmvér je založený na ESP8266 NONOS SDK 1.4.0. Z poskytnutých zdrojových kódov si môžeme skompilovať vlastnú verziu a pridať alebo odobrať softvérové moduly, ktoré budú prítomné ako napr. PWM, časovač, práca so súbormi a ďalšie. [4]

#### 3.3.2 Vlastný firmvér

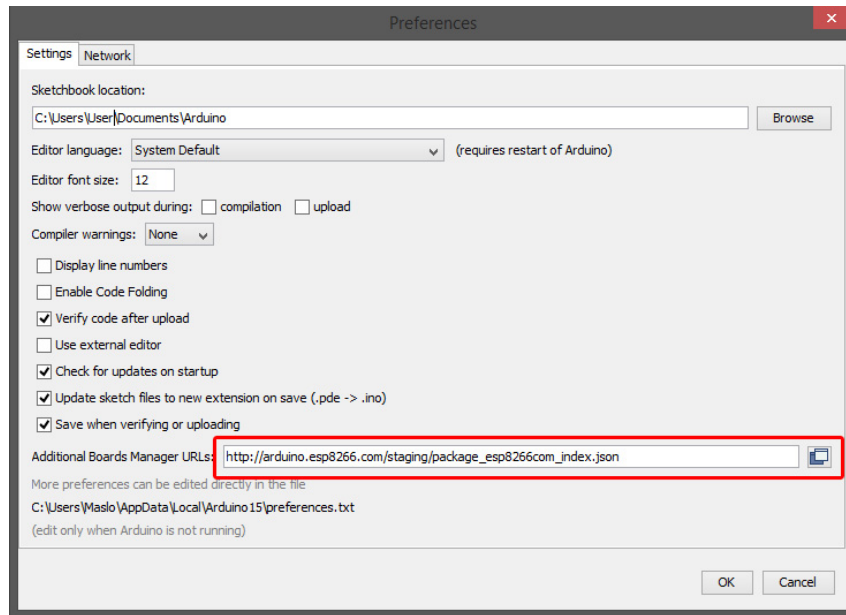
V prípade, že chceme vytvoriť vlastný firmvér máme viacero možností.

Jednou z nich je pri programovaní použiť priamo Espressif SDK. Čo vyžaduje nainštalovať vývojové prostredie s podporou jazyku C ako napríklad Visual Studio alebo Eclipse. Nainštalovať a nastaviť Espressif SDK, ESP image tool a ESP upload tool. Podrobný postup akým spôsobom je možné toolchain zostaviť je možné nájsť na tejto adrese:

*<https://github.com/esp8266/esp8266/wiki/wiki/Toolchain>*

### 3.3.3 Arduino core for ESP8266

Najjednoduchšou možnosťou ako vytvoriť vlastný firmvér a upraviť funkčnosť modulu je použitie vývojového prostredia Arduino IDE v spojení s balíkom *Arduino core for ESP8266*. Stačí nainštalovať Arduino IDE 1.6.4 alebo novšie, následne pridať do **Additional Boards Manager URLs**: adresu - [http://arduino.esp8266.com/staging/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/staging/package_esp8266com_index.json) a nastavenia uložiť.



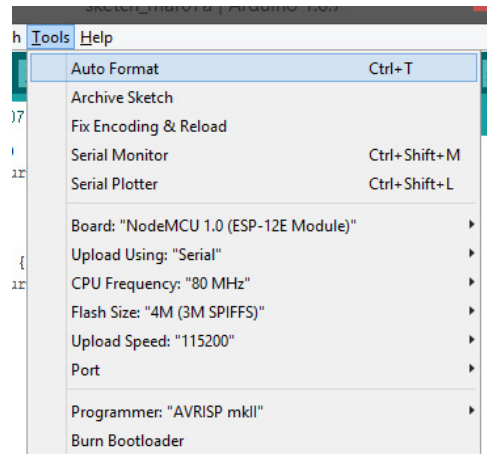
Obr. 12: Nastavenia Arduino IDE

Po pridaní URL je balíček EPS8266 dostupný v Board Manager-y a tu tento balíček vyberieme.



Obr. 13: Board Manager

Poslednými krokmi je už len vybrať správne nastavenia pre nami vybraný modul ako je typ modulu, spôsob komunikácie s modulom, frekvencia procesoru, veľkosť flash pamäte a rýchlosť komunikácie s modulom počas nahrávania firmvéru.



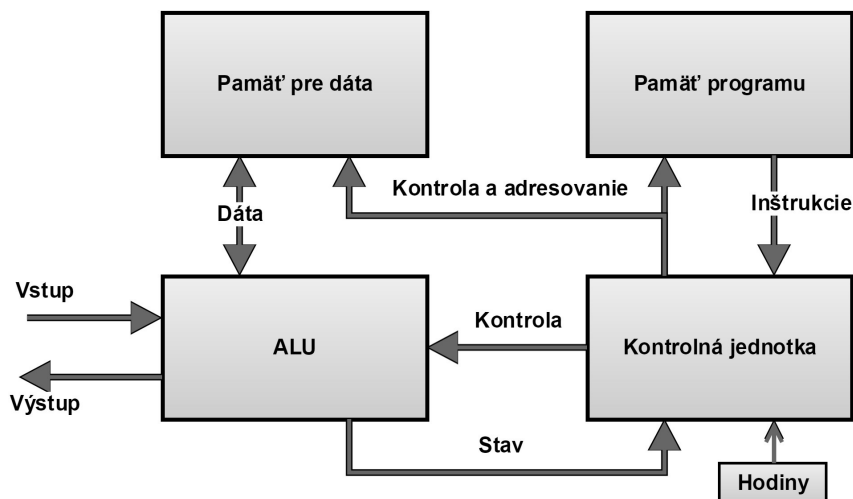
Obr. 14: Nastavenia IDE pre modul

Po týchto troch jednoduchých krokoch je IDE pripravené na použitie s modulom ESP8266 v jeho rôznych prevedeniach. Správanie prostredia a kompilácie kódu je ďalej rovnaká ako v prípade Arduino modulov. Stačí pripojiť modul pomocou sériového rozhrania a nahráť nami vytvorený program.[5]

## 4 Jednočipové mikrokontroléry PIC

Mikrokontroléry PIC sú programovateľné polovodičové súčiastky - jednočipové mikropočítače vyrábané firmou Microchip Technology, odvodené od PIC1650 pôvodne vyvinutej General Instrument's Microelectronics Division. Názov PIC je skratka pre *Peripheral Interface Controller*.

Využívajú sa v domácich spotrebičoch, automobiloch, meracích a riadiacich systémoch. Mikroprocesory rady PIC sú založené na Hardvardskej architektúre.



Obr. 15: Hardvardská architektúra

Výhodou mikrokontrolérov je cena a nízka spotreba. Majú malú sadu inštrukcií, čo prospieva k rýchlemu osvojeniu si znalostí a jednoduchému programovaniu. Vstavaný oscilátor s možnosťou nastavenia frekvencie.

Veľké množstvo periférií vrátane I<sup>2</sup>C, SPI, USB, USART, A/D, PWM, LIN, CAN, PSP, Ethernet. Dostupnosť procesorov vo vyhotovení púzdra typu DIL a relatívne lacnému programátoru PICkit prospieva k cenovo dostupnej a jednoduchšej tvorbe prototypov.

### 4.1 Rada mikrokontrolérov PIC18

Rada mikroprocesorov PIC18 kombinuje maximálnu úroveň výkonu a integrácie spolu s jednoduchosťou 8-bitovej architektúry. S výpočtovým výkonom až 16 MIPS mikroprocesory rady PIC18 používajú rozšírené periférie ako napríklad CAN, USB, Ethernet a LCD.

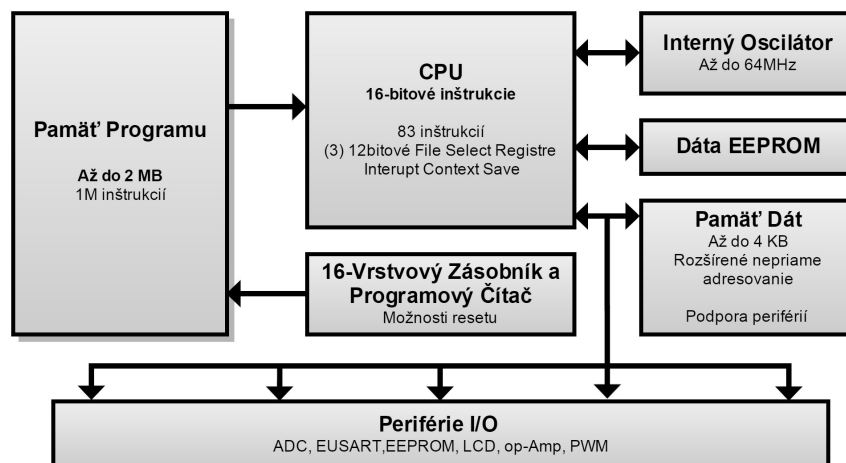
Rada mikroprocesorov PIC18 je rozdelená na dve časti:

- Rada "PIC18J"

určená pre použitie v cenovo citlivých aplikáciách vyžadujúcich veľkú pamäťovú hustotu

- Rada "PIC18K"

poskytuje najvyšší výkon v nanoWatt XLP benchmark-u s extrémne nízkou spotrebou.



Obr. 16: Mikroprocesory PIC18

## 4.2 Vývojové prostredie MPLAB X

Jedná sa o poslednú verziu vývojového prostredia MPLAB IDE postaveného na open-source platforme Netbeans IDE. Zvolením tejto platformy je možné veľmi rýchlo a jednoducho vyvíjať a pridávať funkcie do tohto prostredia. Toto vývojové prostredie ponúka podporu editovania, programovania a ladenia Microchip 8, 16 a 32 bitových PIC mikrokontrolérov.[7]

MPLAB X je prvá verzia, ktorá pridáva multiplatformovú podporu pre Mac OS X a Linux k stávajúcej podpore MS Windows.

MPLAB podporuje nasledujúce kompilátory kódu:

- MPLAB XC8 - C kompilátor pre 8-bitových PIC mikroprocesoroch
- MPLAB XC16 - C kompilátor pre 16-bitových PIC mikroprocesoroch
- MPLAB XC32 - C++ kompilátor pre 32-bitových PIC mikroprocesoroch

Tie kompilátory vychádzajú v dvoch verziách - Free ,Standard a PRO. Tieto verzie sa líšia úrovňou optimalizácie, kde verzia Free má úroveň najnižšiu a verzia PRO má najvyššiu - táto verzie je však platená. [8]

### 4.3 Programátor PICKit 3

PICKit 3 je jednoduchý a lacný programátor / debugger ovládaný PC, na ktorom beží MPLAB X (v8.2 a vyššie). PICKit 3 je integrovaná súčasť vývojárskeho balíku nástrojov. Aplikčné využitie sa však môže líšiť od softvérového vývoju až k hardvérovej integrácii. [6]



Obr. 17: Programátor PICKit 3

PICKit 3 sa využíva na vývoj softvéru a hardvéru pre Microchip PIC mikrokontrolérov a digitálnych signálnych kontrolérov založených na In-Circuit Serial Programming. Taktiež bol vyvinutý na emulovanie procesorov s možnosťou debug-ovania.[6]

PICKit 3 ponúka nasledujúce výhody:

- podporu plnej rýchlosti USB použitím štandardných Windows ovládačov
- debug-ovanie v reálnom čase
- vstavaný monitoring prepätia a skratu
- nízke napätie do 5 V (rozsah 1,8 - 5 V)
- diagnostika za pomoci vstavaných LED (power, active, status)

[6]

## 5 OS android

Android je mobilný operačný systém vyvíjaný firmou Google, je založený na Linuxovom jadre a je primárne určený pre mobilné zariadenia s dotykovou obrazovkou, akými sú mobilné telefóny a tablety. Užívateľské rozhranie Android-u je založené na priamej manipulácii použitím dotykových gest spolu s dotykovou klávesnicou na písanie textu. Google taktiež vyvíja Android TV pre použitie v televíziách, Android Auto pre použitie v autách a Android Wear pre inteligentné hodinky.

Android je celosvetovo dominantný operačný systém pre mobilné zariadenia.[9] Momentálne sa nachádza na miliónoch mobilných zariadení vo viac ako 190 krajinách sveta.

Je vystavaný v spolupráci s open-source Linuxovou komunitou a ďalšími 300 partnermi v oblasti hardvéru, softvéru a poskytovania telekomunikačných služieb. Vďaka tomu sa stal Android najrýchlejšie sa rozvíjajúcim mobilným OS.

### 5.1 Architektúra

Architektúra OS Android by sa dala rozdeliť na viacero vrstiev. Každá z nich má svoj účel a nemusí byť oddelená od ostatných.



Obr. 18: Architektúra Andorid OS

Vrstvy môžeme rozdeliť nasledovne:

- Aplikácie - napísaná v Jave, bežia na virtuálnom stroji Dalvik (do verzie 4.4) / Android Runtime (Android 5+)
- Framework-ové služby a knižnice - väčšinou písané v Jave - väčšina tohoto kódu beží vo virtuálnom stroji
- Natívne knižnice, démoni a služby - písane v C a C++
- Linuxové jadro - obsahuje ovládače pre hardvér, sieťovú komunikáciu, prístup k súborovému systému a vnútro-procesovú komunikáciu.

[10]

## 5.2 Vývoj aplikácií

Android poskytuje rozsiahly aplikačný framework, ktorý dovoľuje vytvárať inovatívne aplikácie a hry pre mobilné zariadenia za pomoci programového jazyka Java.

Za pomoci Android SDK nástrojov sa zdrojový kód kompiluje spolu so všetkými dátami a súbormi do *Android Package* - archivačný súbor s príponou *.apk*. Tento súbor obsahuje všetko potrebné na nainštalovanie aplikácie na mobilné zariadenie.

Po inštalácii beží každá aplikácia z dôvodu bezpečnosti v izolovanom prostredí - *sandbox*:

- Android OS je v podstate Linuxový multi-user systém, kde každá aplikácia je samostatný užívateľ
- V základe priradí systém každej aplikácii unikátne Linux user ID (ID je používané iba systémom a aplikácií nieje známe). Systém následne nastaví povolenia pre všetky súbory aplikácie takže len ID priradené aplikácii môže k súborom pristupovať.
- Každý proces beží na vlastnom virtuálnom stroji, takže aplikácie sú od seba izolované.
- Všetky aplikácie sa spúšťajú vo vlastnom Linuxovom procese. Operačný systém spustí aplikáciu, keď nejaký komponent aplikácie musí byť spustený. Proces je ukončený v momente kedy už nieje potrebný alebo keď systém potrebuje uvoľniť pamäť.

Týmto spôsobom operačný systém implementuje *princíp najnižších privilégií*. To znamená, že každá aplikácia má prístup iba ku komponentom, ktoré potrebuje na správny beh a k ničomu inému. Toto vytvára veľmi bezpečné prostredie, v ktorom aplikácie nemôžu pristupovať k časti systému, na ktoré nemajú potrebné povolenia.



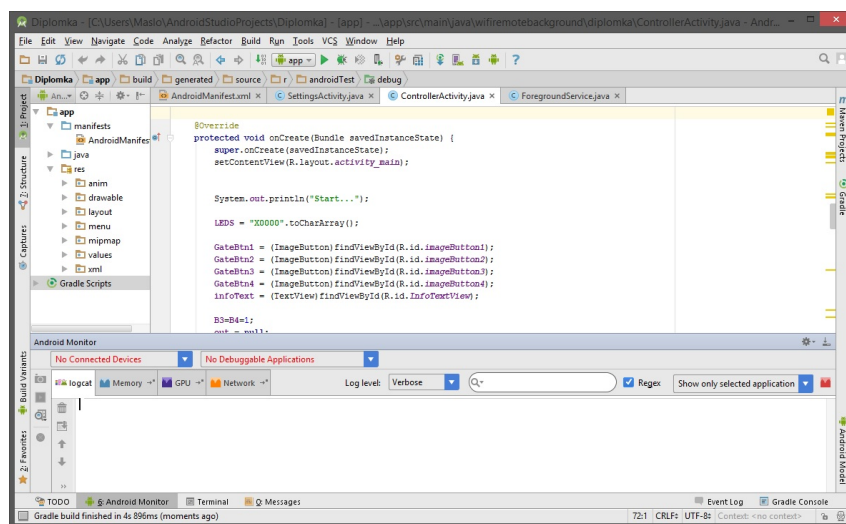
Samozrejme existuje možnosť ako zdieľať dáta medzi aplikáciami a systémovými službami:

- Je možné použiť jedno Užívateľské ID pre dve aplikácie, týmto spôsob aplikácie môžu vzájomne pristupovať k svojim súborom. Pokiaľ dve aplikácie zdieľajú jedno user ID je možné ich spustiť na jednom virtuálnom stroji a aj v tom istom Linuxovom procese, avšak musia byť podpísané tým istým certifikátom.
- Aplikácia môže požiadať o povolenia na prístup k dátam ako užívateľské kontakty, SMS správy, SD karta, fotoaparát, Bluetooth, a ďalším. Tieto povolenia musia byť potvrdené užívateľom - pri inštalácii aplikácie.

[11]

### 5.3 Android Studio

Android Studio je oficiálnym IDE pre vývoj aplikácií určených pre operačný systém Android, založené na IntelliJ IDEA.



Obr. 19: Android Studio IDE

Navyše k výkonnému IntelliJ editoru a vývojovým nástrojom, Android Studio ponúka viacero funkcií, ktoré zvyšujú produktivitu pri vývoji aplikácií, ako napríklad:

- Flexibilný zostavovací systém založený na *Gradle*
- Šablóny na pomoc pri tvorbe základných prvkov aplikácie
- Rozsiahly *Drag&Drop* editor grafického rozhrania
- Nástroj Lint, ktorý zachytáva chyby v zdrojovom kóde, ktoré by mohli spôsobiť problémy s výkonom, kompatibilitou a použiteľnosťou aplikácie

- Zmenšovanie kódu s *ProGuard* a zdrojov s *Gradle*
- Vstavaná podpora pre *Google Cloud Platform*

Pred oficiálnym vydaním Android Studio spoločnosťou Google v Decembri 2014, sa na vývoj aplikácií pre Android oficiálne používalo vývojové prostredie Eclipse + ADT(Android Development Tool). Projekty vytvorené v Eclipse je možné exportovať a následne importovať do Android Studio IDE.

Výhody Android Studio oproti Eclipse + ADK sú hlavne rýchlosť, menšie nároky na výkon, lepší našepkávač, rýchla a jednoduchá inštalácia. Nevýhodami je menší počet podporovaných pluginov, nemožnosť vytvárať projekty pre iné platformy ako Android a tiež nie je možné pri tvorbe projektu použiť iný systém ako Gradle.

[12]

## 6 Realizácia modulu

V tejto časti práce bude popísaná realizácia modulu pre inteligentný dom iDŕm. Jedná sa hlavne o výber vhodných komponentov, návrh zapojenia a programovanie jednotlivých častí. Následne realizácia aplikácie pre OS Android, ktorá bude s modulom komunikovať.

### 6.1 Návrh modulu

Pri návrhu modulu bolo treba zvoliť komponenty, ktoré sú schopné zaistiť plnohodnotnú funkčnosť modulu a poskytujú dostatočné množstvo periférií aby boli zaistené nasledujúce podmienky.

- dostupnosť pomocou Wi-Fi / funkčnosť v režime prístupového bodu
- modul by mal byť schopný rozpoznať klientov, ktorý sa pripoja a odoslať MAC adresu do systému iDŕm
- funkčnosť modulu musí byť zaistená v prípade použitia modulu samostatne ako aj v spolupráci so systémom iDŕm
- možnosť spínať výstupy a poskytovať spätnú väzbu na základe vstupov modulu

Prvou časťou návrhu bol výber vhodného Wi-Fi modulu schopného zaistiť spojenie s mobilným telefónom. Pri hľadaní vhodného kandidáta som narazil na modul ESP8266 navrhnutý firmou Espressif. Tento modul sa vyznačuje relatívne veľkým výkonom, nízkou cenou čím si získal podporu komunity, čo vyústilo k rôznym možnostiam programovania a využitia tohto modulu (viac kapitola 3).

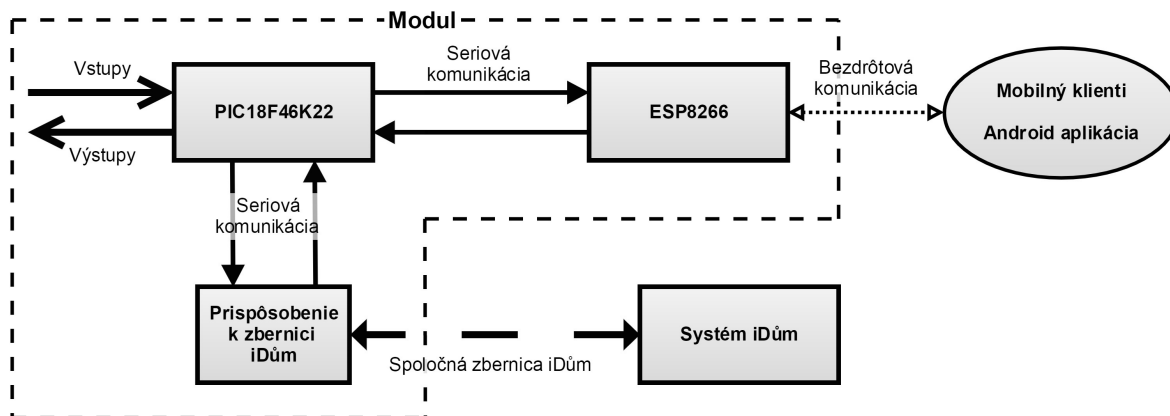
Ďalšou časťou návrhu bola voľba správneho mikroprocesoru ako hlavnú časť modulu, ktorá bude mať na starosť komunikáciu so systémom iDŕm, spínanie výstupov a snímanie zmien na vstupoch. Vhodnou voľbou boli mikrokontroléry firmy Microchip. Vzhľadom k ich spoľahlivosti a tomu, že viacero častí systému iDŕm pracuje na týchto mikrokontroléroch. Pre potrebu dvoch sériových portov, jeden na komunikáciu z Wi-Fi modulom a druhý so systémom iDŕm, množstvo portov na spínanie výstupov a snímanie vstupov bol zvolený PIC18F46K22 (viac kapitola 4).

Keďže prototyp je realizovaný na nepájivom poli, boli na reprezentáciu výstupov použité LED diódy a tlačidlá ako náhrada za vstupy resp. spätnú väzbu. Na pripojenie k zbernici systému iDŕm bol vytvorený prispôbovací obvod.

## 6.2 Logické zapojenie

Logické zapojenie modulu je znázornené na obrázku 21, modul ESP8266 komunikuje s pripojenými klientmi za pomoci bezdrôtového pripojenia Wi-Fi. Klienti sa pripájajú pomocou aplikácie ako k bežnému prístupovému bodu.

Medzi ESP8266 a mikrokontrolérom PIC18F46K22 prebieha komunikácia zabezpečená pomocou sériového / UART rozhrania.



Obr. 20: Logické zapojenie modulu

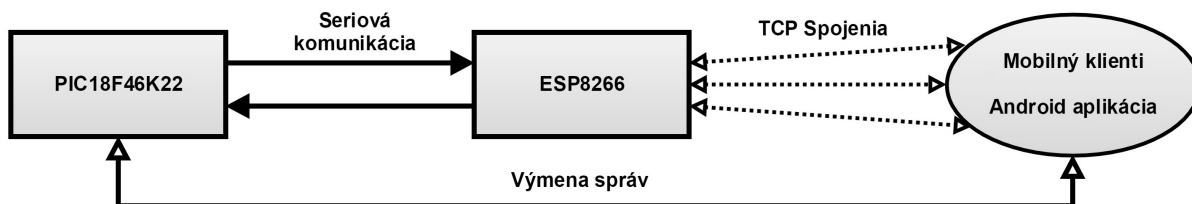
Komunikácia medzi mikroprocesorom a systémom iDŭm je taktiež založená na sériovom prenose avšak na rozdiel od štandardného sériového prenosu, kde sa na prenos každým smerom používa samostatný vodič je v tomto prípade použitá zbernica obsahujúca len jeden dátový vodič. Z toho vyplýva, že na pripojenie mikrokontroléra k systému je nutný prispôbovací obvod, ktorý nám dvojvodičovú komunikáciu prispôsobí na jednovodičovú.

Všetky periférie ako napr. výstupy reprezentované LED diódami a vstupy reprezentované tlačidlami sú pripojené k mikrokontroléru.

### 6.2.1 Komunikačné protokoly

Komunikačný protokol medzi mikrokontrolérom a systémom iDŭm je popísaný v kapitole 2.2.2, táto časť bude skôr venovaná popisu komunikácie medzi mobilnými klientmi a ESP8266, a následne medzi ESP8266 a mikroprocesorom PIC.

Mobilný klient sa pripojí k prístupovému bodu vysielaného Wi-Fi modulom ESP8266. Na module beží DHCP server, ktorý priradí modulu IP adresu z naprogramovaného rozsahu. Po pripojení k modulu a získaní IP si aplikácia vytvorí TCP socket a pokúsi sa nadviazať spojenie s Wi-Fi modulom. Po úspešnom nadviazaní spojenia môže prebiehať komunikácia.



Obr. 21: Popis komunikácie medzi jednotlivými prvkami

Všetky dáta prijaté zo sériového rozhrania sú kopírované znak po znaku a posielané všetkým pripojeným klientom. Následne všetky správy od klientov prijaté cez TCP spojenie sú posielané na sériové rozhranie smerom ku mikrokontrolérovi.

Medzi aplikáciou a mikrokontrolérom sa posielajú 3 typy správ:

1. reprezentácia stlačených tlačidiel v aplikácii

Je posielaná formou reťazca "X1111", kde prvý znak "X" je identifikáciou, že sa jedná o stav tlačidiel a následne binárne znázornená reprezentácia stavu jednotlivých tlačidiel

2. spätná väzba z modulu reprezentujúca zmenu na vstupe mikroprocesoru

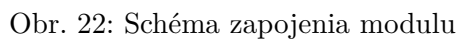
Podobne ako predchádzajúci reťazec "S1111", kde znak "S" reprezentuje spätnú väzbu a nasledujúce 4 znaky reprezentujú logický stav vstupov na mikrokontroléry

3. text, ktorý sa zobrazí na displeji mobilného telefónu

Správa určená na zobrazenie na displeji mobilného telefónu "#text" začína znakom "#", ktorý definuje tento typ správy a nasleduje text, ktorý sa už zobrazí na displeji

Koniec správy je vždy signalizovaný koncom riadku, znak "/n".

V nasledujúcej kapitole bude zobrazené fyzické zapojenie jednotlivých častí obvodu. Obrázok 22 zobrazuje kompletne zapojenie, ktoré bolo následne realizované na nepájivom poli. Zobrazenom na obr. 23.

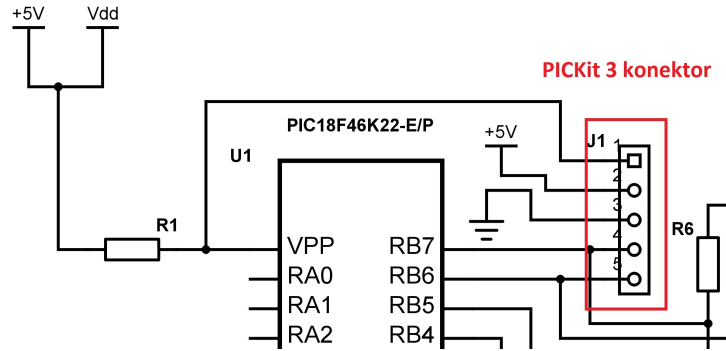


Obr. 23: Realizácia modulu na nepájivom poli

### 6.3.1 Zapojenie mikrokontroléra

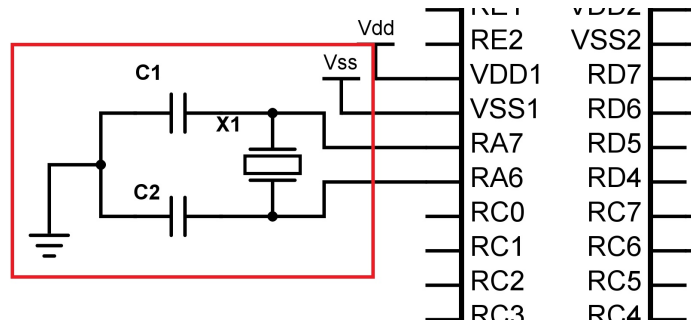
Mikrokontrolér ako aj celý modul je napájaný 5 V. Oba piny VDD1/VDD2 sú pripojené na +5 V. Piny VSS1/VSS2 sú pripojené ku GND.

Na obrázku 24 vidíme spôsob zapojenia konektora použitého na programovanie mikrokontroléra.



Obr. 24: Konektor pre programovanie mikrokontroléra

Pre nastavenie pracovnej frekvencie bol použitý 16 MHz kryštál(X1), spoločne s paralelným zapojením dvoch keramických kondenzátorov (C1, C2) obr. 25.



Obr. 25: Konektor pre programovanie mikrokontroléra

Kapacita kondenzátorov bola odvodená podľa rovnice:

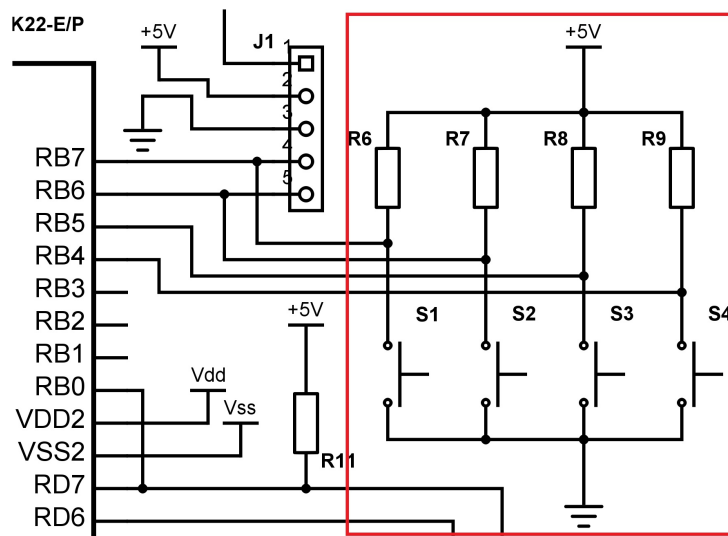
$$C_{krys} = ((C1 \times C2)/(C1 + C2)) + C_{par} \quad (1)$$

Kde  $C_{krys}$  je zaťažovacia kapacita kryštálu, v našom prípade 9pF a  $C_{par}$  je parazitná kapacita zapojenia, môže byť v rozsahu 2-5 pF.

$$C1 = C2 = 2(C_{krys} - C_{par}) = 2(9 - 2.5) = 15pF \quad (2)$$

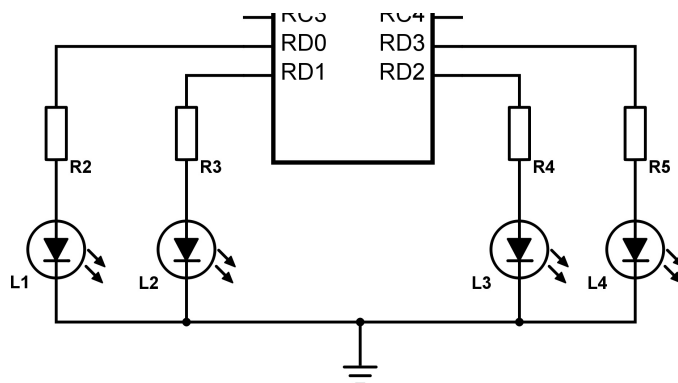
Po zjednodušení rovnice nám vyšla kapacita 15pF.

Pre reprezentáciu spätnej väzby boli použité tlačidlá. V základnej polohe sú pripojené na napätie +5 V pomocou 10 kΩ rezistoru. Po stlačení tlačidla, je na vstupný pin privedená log. 0.



Obr. 26: Spätná väzba reprezentovaná tlačidlami

Pre názorne zobrazenie ovládania výstupov mikrokontroléru boli použité LED diódy, zapojené podľa obr. 27. Diódy sú v sériovom zapojení spolu so 100  $\Omega$  rezistorom, ktorý zaisťuje prúdové obmedzenie diódy na 35 mA.



Obr. 27: Výstupy reprezentované LED diódami

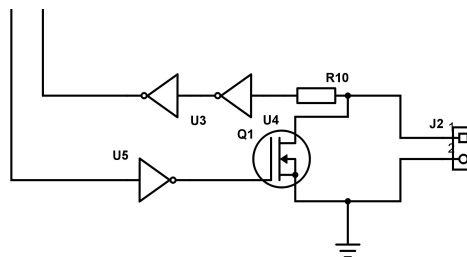
Wi-Fi modul ESP je pripojený k mikrokontroléru pomocou dvoch vodičov, RC7-Rx a RC6-Tx. Mimo napájanie Wi-Fi modulu sa jedná o jediné fyzické pripojenie modulu.

### 6.3.2 Konfigurátor iDŭm

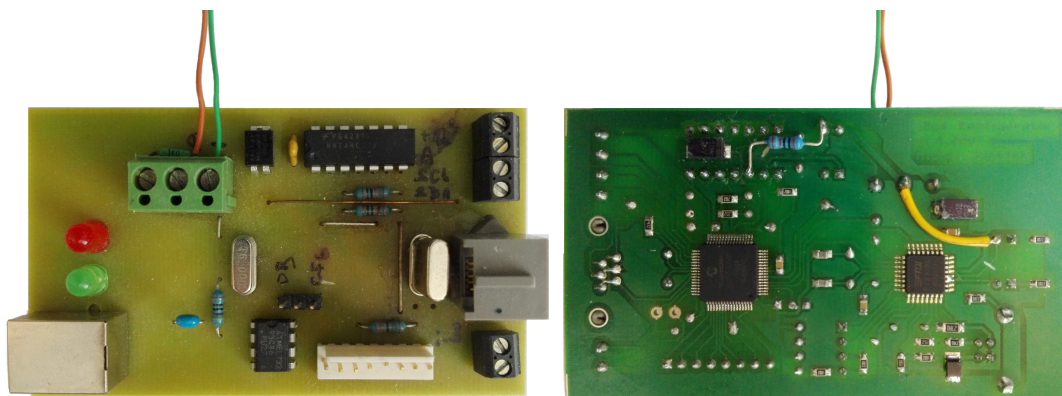
Pripojenie k spoločnej zbernici systému iDŭm je realizované pomocou prispôbovacieho obvodu, obr. 28. Pomocou obvodu sa dvojvodičová komunikácia prispôsobí na komunikáciu pomocou jednovodičovej zbernice.

Tento obvod je pripojený k pinom RD7-Rx spolu s RB0, a Tx k pinu RD6 mikrokontroléra. Kompletný popis a spôsob fungovania zbernice je v podkapitole 2.2.1.





Obr. 28: Prispôsobenie sériovej komunikácie pre jednovodičovú zbernicu



Obr. 29: iDum Konfigurátor

Pri overovaní funkčnosti modulu a schopnosti komunikácie so systémom iDum bol použitý iDum Konfigurátor. Na obr. 29 vidíme tento konfigurátor a vyvedenú zbernicu. Zelený vodič predstavuje spoločnú zem a oranžový vodič dátovú zbernicu.

## 6.4 Program mikrokontroléra PIC

Nasledujúca podkapitola bude popisovať jednotlivé časti programu vyvinutého pre mikrokontrolér PIC. Na programovanie a ladenie programu bol použitý MPLAB X IDE v3.20, kompilátor XC8 a PICKit 3.

### 6.4.1 Prerušená

Počas behu programu sa môže objaviť viacero prerušení, či už sa jedná o vnútorné prerušená alebo prerušená po zmene stavu vstupov.

- Časovač TMR1

Jedným z najdôležitejších je prerušenie od časovača TMR1. Toto prerušenie je nastavené tak aby sa objavilo presne každú 1 ms. Hlavnou funkciou je meranie času, ktorý sa následne používa v ďalších častiach programu. Ako sledovanie času zopnutia výstupov a sledovanie jednotlivých časových úsekov potrebných pre komunikáciu na zbernici systému iDum.

- **Zmena stavu vstupov**

V prípade zmeny vstupu či už z nízkej úrovne na vysokú alebo naopak sa objaví prerušenie. Nastaví sa čas zákmitu tlačidla, počas ktorého sa nebude sledovať zmena na vstupoch. Tento čas sa každým prerušením časovača TMR1 znižuje o jedna a v momente dosiahnutia nuly je sledovanie zmien na vstupoch obnovené a prerušenie sa po zmene môže znovu objaviť. Taktiež sa nastaví príznakový bit *BylaZmenaVypinace* a stav tlačidiel je vyhodnotený v hlavnej funkcii.

- **Prichádzajúca správa zo systému iDúm**

Prerušenie spôsobené prichádzajúcim znakom správy, nastaví register časovaču TMR1 na odpočet 1 ms a nastaví sa príznakové bity upozorňujúce na prijímanie správy. Prichádzajúca správa sa zapisuje do *dočasnej* správy znak po znaku až po posledný, kde sa spočíta CRC a vyhodnotí sa či prišla správa neporušená. Pri prijatí tretieho znaku správy (*DevID*) sa vyhodnotí, či je správa určená pre modul. Ak je správa určená pre modul pokračuje sa v prijímaní dát, ak nieje v prijímaní dát sa pokračuje ale tieto dáta niesu na konci prenosu spracované. Po prijatí celej správy sa nastaví príznakový bit *ZpravaPrijata* a správa sa vyhodnotí v hlavnej funkcii.

- **Prichádzajúca správa z Wi-Fi modulu**

Prerušenie vyvolané prichádzajúcim znakom správy z Wi-Fi modulu na sériové rozhranie mikrokontroléra, nastaví register časovaču TMR1 na odpočet 1 ms. Prijatý znak sa uloží do zásobníka a hodnota pozície správy sa zvýši o jedna. Toto sa opakuje pokiaľ sa neobjaví znak `"\n"`, ktorý predstavuje koniec správy.

Správa sa následne uloží do *dočasnej* premennej, vynuluje sa pozícia v správe, vyčistí sa zásobník a aj registre USART rozhrania, čím sa pripraví na ďalšiu správu. Po prijatí kompletnej správy sa nastaví príznak *ZpravaPrijata Wi-Fi* a správa sa vyhodnotí v ďalšom cykle hlavnej funkcie.

#### 6.4.2 Hlavná funkcia

Hlavnú funkciu môžeme rozdeliť na dve časti, prvou je inicializačná časť. V tej to časti sa nastavujú registre, ktoré určujú funkciu jednotlivých pinov mikrokontroléru, či už sa jedná o nastavenie výstupov pre LED diódy, vstupov pre tlačidlá tak aj nastavenie pinov pre správnu funkčnosť sériových rozhraní. Nastavuje sa tu aj typ prerušenia, inicializácie správnej rýchlosti pre jednotlivé sériové rozhrania, vynulovanie príznakových bitov a načítanie premenných z EEPROM pamäte.

Druhou časťou programu je nekonečný cyklus. V základnom stave, kde všetky príznakové bity sú vynulované beží tento cyklus naprázdno a nevykonáva žiadne akcie. V prípade, že sa objaví jedno z možných prerušení a nastaví niektorý z príznakových bitov a v ďalšom cykle sa vykoná časť programu závislá na type príznaku.

Môžu sa objaviť nasledujúce príznaky:

- **Zmena stavu vstupov**

V prípade nastavenia príznakového znaku pri zmene stavu vstupov sa v prvom rade porovná či ide o prerušenie vyvolané povoleným tlačidlom (*v konfigurácii sa dá nastaviť ktoré tlačidlá budú povolené, resp. zmena ktorých sa bude vyhodnocovať*).

V prípade, že zmena bola vyvolaná povoleným tlačidlom, aktuálny stav tlačidiel sa uloží do premennej a ak je definovaný ID riadiaceho počítača systému je stav vyslaný do systému iDům. V opačnom prípade sa stav vysielá priamo do Wi-Fi modulu.

- **Správa prijatá zo systému iDům**

Prijatá správa nastaví príznakový bit, vďaka ktorému sa spustí spracovanie správy v hlavnej funkcii. Prichádzajúce správy sú rozdelené na správy typu Byte, Pole a konfiguračná správa.

#### **Správa Byte**

Ak je správa typu Byte, zisťuje sa pre koho presne je správa určená "*UserID*".

V prípade, že je správa určená pre Wi-Fi modul (*UserID = 0x11*) a tým pádom pre pripojených klientov je správa transformovaná do reťazca znakov, príklad :

*Prichádzajúci bajt má hodnotu 0x0a - binárne 1010 - táto hodnota je transformovaná do reťazca "S1010\n\0" a vyslaná na sériové rozhranie smerom k Wi-Fi modulu.*

Správa, ktorá má *UserID=0* je priamo spracovaná, binárne vyjadrenie posledných štyroch bitov určí, ktoré výstupy budú zopnuté na určitý čas. V momente kedy sa zopne daný výstup sa nastaví čas po akú dobu má byť tento výstup zopnutý - tento čas je v konfigurácii nastavený pre každý výstup samostatne, príklad:

*Bajt má hodnotu 0x0a - binárne 1010 - výstupy 1 a 3 sa zopnú na čas daný v konfigurácii*

Čas pre každý výstup sa znižuje pri prerušení spôsobenom časovačom TMR1.

#### **Správa Pole**

Pokiaľ je prichádzajúca správa typu pole, porovná sa či je určená pre Wi-Fi modul (*UserID = 0x11*).

V správe je obsiahnutý text, ktorý sa má zobrazíť v mobilnej aplikácii. Tento text sa po prijatí upraví a následne prepošle na sériové rozhranie smerom k Wi-Fi modulu.

Prijatý text od systému iDům - test - sa upraví na "#test\n\0"

#### **Konfiguračné správy**

Konfiguračné správy slúžia na zmenu konfigurácie mikrokontroléra. Táto konfigurácia je uložená v EEPROM pamäti. Spracovanie konfiguračných správ je takmer totožné so spracovaním použitým v iných moduloch pre systém iDům.

Správy, ktoré patria do typu konfiguračných správ sa ďalej rozdeľujú na:

#### **Inicializáciu konfigurácie**

Táto správa musí byť doručená ako prvá pred ostatnými konfiguračnými správami.

#### **Načítanie pamäte EEPROM**

V prípade prijatia tejto správy je načítaný obsah EEPROM pamäte vložený do správy a odoslaný do systému iDŕm.

#### **Zápis do pamäte EEPROM**

Prijatá správa obsahuje dáta, ktoré budú uložené do EEPROM pamäte mikrokontroléra, týmto spôsobom sa zmení konfigurácia a správanie mikrokontroléra.

#### **Reset mikrokontroléra**

Po prijatí správy sa mikrokontrolér reštartuje a znovu inicializuje.

#### **Získanie ID mikroprocesoru**

Odpoveďou na túto správu je vyslanie ID čísla.

#### **Ukončenie konfigurácie**

Správa ukončí konfiguračný režim mikrokontroléra a ten následne nevykonáva vyššie uvedené správy, kým konfigurácia nie je znovu inicializovaná.

Mikrokontrolér odpovedá na všetky tieto správy pozitívne správou cfgOK alebo negatívne cfgFailed.

### **• Správa prijatá z Wi-Fi modulu**

Z Wi-Fi modulu a teda od klientov pripojených môžu prísť 2 druhy správ:

#### **Stlačenie tlačidla**

V prípade že, dôjde k stlačeniu tlačidla v aplikácii, je táto zmena vyslaná formou reťazca z aplikácie na Wi-Fi modul a preposlaná do mikrokontroléra. Správa ma formát “X1010”, kde X informuje, že sa jedná o zmenu stavu tlačidiel a nasledujúce 4 čísla určujú, ktoré tlačidlá majú akú logickú hodnotu. 1 reprezentuje že tlačidlo bolo stlačené - 0 že stlačené nieje.

X1000 - tlačidlo číslo štyri bolo stlačené.

#### **MAC adresa klienta**

SDK pre Wi-Fi modulu nemá implantovanú prácu s MAC adresami, ale pokiaľ je tento modul spustený v debug režime posiela MAC adresy pripojených klientov na sériové rozhranie pri pripojení ako aj odpojení klienta.

Na tieto správy mikrokontrolér reaguje - ak sa v správe nachádza reťazec “station” - jedná sa o pripojenie alebo odpojenie klienta. V prípade že, sa klient pripojil je v správe obsiahnutý reťazec “join” ak klient sieť opustil je reťazcom “leave”.

Tieto reťazce sú vyhľadávané v správe a podľa toho sú formulované do správy pre systém iDŭm.

- **Čas zopnutia výstupu vypršal**

Každý výstup má nastavený presný čas, po ktorý má byť zopnutý. Po zopnutí toho výstupu sa začne čas odpočítavať. V tejto časti programu sa čaká kým premenná CasLedx, kde “x” je číslo výstupu, dosiahne hodnotu 0. V tomto prípade je výstup opäť uvedený do vypnutého stavu.

#### 6.4.3 Pamäť EEPROM

Pamäť EEPROM slúži na uloženie konfigurácie mikrokontroléra, na konfiguráciu je použitých len 8 bajtov z celkových 1024 dostupných.

Konfigurácia je uložená nasledovne:

- Byte 0 - ID číslo mikrokontroléra

Určuje adresu mikrokontroléra na zbernici systému iDŭm.

- Byte 1 - Čas zákmitu tlačidla

Po stlačení tlačidla vzniká zákmit, ktorý spôsobí viacnásobnú zmenu logickej úrovne na vstupe. Na ošetrenie tohto problému je určená doba, po ktorú mikrokontrolér nebude reagovať na zmenu na vstupe po prvom stlačení tlačidla.

- Byte 2 až 5 - Čas zopnutia pre každý zo štyroch výstupov

Je možné nastaviť dobu, po ktorú bude výstup zopnutý, túto dobu je možné nastaviť pre každý zo štyroch výstupov.

- Byte 6 - Povolené

Nastavuje, na ktoré vstupy má mikrokontrolér reagovať respektíve, pri ktorých vstupoch má o zmene informovať systém iDŭm / mobilnú aplikáciu.

- Byte 7 - ID číslo riadiaceho počítača systému iDŭm

Číslo určujúce adresu riadiaceho počítača systému iDŭm. Hodnota 0 znamená, že je modul používaný bez pripojenia k systému iDŭm.

Na zápis a čítanie z EEPROM som musel vytvoriť vlastné metódy podľa datasheet-u pre PIC18K46F22, pretože metódy implementované v knižnici pre kompilátor XC8 nie sú podporované resp. nedokázali prístup k EEPROM pamäti a načítať alebo zmeniť jej obsah.

## 6.5 Program Wi-Fi modulu ESP8266

Program pre Wi-Fi modul je veľmi jednoduchý, zložený z dvoch funkcií. Na programovanie bolo použité Arduino IDE + Arduino core for ESP8266.

Štandardne nastavený názov prístupového bodu je “**Gate**” a použité heslo “**12345678**”

- **Inicializačná funkcia**

Je spustená pri zapnutí modulu a nastavuje sa v nej základné parametre ako:

- rýchlosť sériovej linky - 9600 bit/s
- režim Wi-Fi modulu - prístupový bod
- SSID a heslo
- nastavenie TCP serveru
- spustenie debug režimu

- **Hlavná funkcia**

Hlavná funkcia je nekonečná slučka, v ktorej sa obsluhujú jednotlivý pripojený klienti. Ak žiadni klienti nenadviazali TCP spojenie so serverom bežiacim na module, cyklus sa zastaví na funkcii *server.available()*, ktorá čaká kým sa klient pripojí.

Pri pripojení klienta sa hlavná funkcia rozbehne. Rozsvieti sa kontrolná LED dióda, ktorá upozorňuje, že je klient pripojený. Nasleduje cyklus, v ktorom sa postupne prejde pole pripojených klientov a zistí sa či od nich prišla správa, ak správa prišla je preposlaná na sériové rozhranie.

Následne sa prečíta zásobník na sériovej linke, ak sa tam nachádza správa je pomocou cyklu opäť preposlaná všetkým pripojeným klientom.

V momente, keď sa odpojí posledný klient je splnená podmienka, po ktorej dióda na module zhasne - to indikuje, že žiaden klient už nieje pripojený na server a čaká sa na pripojenie nového klienta.

## 6.6 Aplikácia pre OS Android

Aplikácia pre operačný systém android bola navrhnutá tak, aby bola jednoduchá na použitie a ponúkala okamžitý prístup k ovládaniu ak sa užívateľ nachádza v dosahu modulu. Na vytvorenie aplikácie bolo použité Android Studio a aplikácia je kompatibilná s verziami systému 4.4 - KitKat a vyššie.

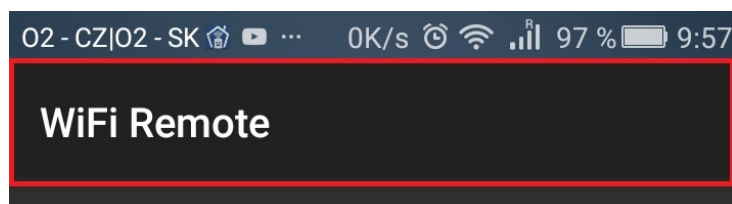
Aplikácia pre obsluhu modulu je zložená z troch častí: Konfiguračná obrazovka, služba na pozadí, obslužná časť.

### 6.6.1 Konfiguračná obrazovka

Táto obrazovka je zobrazená ako prvá po spustení aplikácie. Jej účel je počiatočné nastavenie SSID modulu a hesla. Na tejto obrazovke sa dá taktiež spustiť alebo zastaviť služba na pozadí, ktorá sníma či sa v okolí nachádza modul.

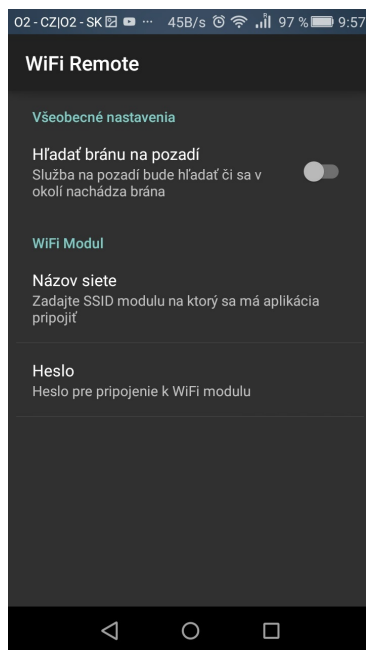
Trieda `SettingsActivity` rozširuje triedu `AppCompatActivity`, ktorá rozširuje triedu `PreferenceActivity`.

`PreferenceActivity` je trieda obsiahnutá v SDK pre Android a je určená na vytváranie nastavení aplikácie. A však neobsahuje `AppCompatActivity` menu (obr. 30), preto bola táto trieda rozšírená triedou `AppCompatActivity`.



Obr. 30: `AppCompatActivity` menu

Trieda `AppCompatActivity` implementuje nevyhnutné volania prvku `AppCompatActivity`. Následne bola táto trieda rozšírená triedou `SettingsActivity`, ktorá obsahuje priamo implementáciu nastavení aplikácie.



Obr. 31: Konfiguračná obrazovka

Obrazovka sa skladá z troch prvkov:

- Spínač - Hľadať bránu na pozadí

Zobrazuje aktuálny stav vyhľadávania modulu na pozadí - zapnuté/vypnuté

- Tlačidlo - Názov siete

Po stlačení tlačidla je vyvolané dialógové okno, v ktorom je možné nastaviť SSID modulom, na ktorý sa snažíme pripojiť

- Tlačidlo - Heslo

Po stlačení tlačidla je vyvolané dialógové okno, v ktorom je možné nastaviť heslo k pripojeniu na modul

Všetky nastavenia sa ukladajú do *SharedPreferences* a nastavenia sú z tohto miesta nahrávané pri spustení aplikácie.

V tejto triede je implementované sledovanie zmien v *SharedPreferences*. Pri zmene prepínača sa vyhodnocuje či je tlačidlo zapnuté alebo vypnuté. Ak je tlačidlo zapnuté služba na pozadí sa spustí a v prípade vypnutia je služba zastavená.

### 6.6.2 Služba na pozadí

Úlohou tejto služby je v určitých intervaloch sledovať aké Wi-Fi siete sa nachádzajú v dosahu a či sa medzi nimi nachádza SSID nastavené pre modul. V prípade, že je modul v dosahu má služba zabezpečiť automatické pripojenie a spustenie obslužnej časti. V momente kedy sa pripojí telefón k modulu sa pozastaví vyhľadávanie modulu a sledujú sa zmeny Wi-Fi pripojenia. Ak sa telefón odpojí od modulu je opäť obnovené vyhľadávanie.

Popis tejto služby by som rozdelil do troch častí:

- **Inicializácia**

Objekty a premenné sa inicializujú v dvoch metódach a to *onCreate* a *onStartCommand*. Tieto metódy prekrývajú pôvodne funkcie rodičovskej triedy.

#### **onCreate**

V metóde sa inicializujú *SharedPreferences* - načíta sa nastavené SSID a heslo Wi-Fi modulu nastavené v konfiguračnej časti, inicializuje sa *Wi-FiManager* potrebný pre skenovanie a prácu z Wi-Fi, vytvorí sa objekty dvoch vnorených tried:

*Wi-FiScanReceiver* - slúži na vyhľadávanie Wi-Fi modulu v okolí

*Wi-FiStatusChange* - sleduje zmeny stavu Wi-Fi pripojenia

Taktiež je vytvorený odkaz na *NotificationManager*, ktorý sa používa na zobrazenie informačných notifikácií alebo ich nasledovnú úpravu.



### **onStartCommand**

Metóda je spustená po zavolaní akcie *start* na objekt našej služby. Ako prvé je vyvolané dialógové okno, ktoré informuje že služba bola spustená a vytvorí sa notifikácia, ktorá zobrazuje aktuálny stav služby.

V prípade, že je mobilný telefón asociovaný s prístupovým bodom je automaticky spustená obslužná časť aplikácie a na pozadí sa spustí sledovanie zmien stavu Wi-Fi. Ak asociovaný nieje, spustí sa skenovanie okolia a hľadanie nami nastaveného prístupového bodu.

#### • **Skenovanie stavu Wi-Fi**

V triede služby sú definované dve vnorené triedy, obe sú rozšírením triedy *BroadcastReceiver*. Tieto dve triedy slúžia na sledovanie stavu Wi-Fi:

##### **Wi-FiScanReceiver**

Pri spustení skenovania sietí je nastavený časovač, ktorý každých 5 sekúnd pošle žiadosť na *Wi-FiManager* o poskytnutie zoznamu sietí. Odpoveď na túto žiadosť je zachytená objektom triedy *Wi-FiScanReceiver*. Tu je zoznam sietí postupne prejdenný a hľadá sa v ňom nami požadované SSID. Pokiaľ zoznam neobsahuje našu sieť nič sa nevykoná.

Ak sa v liste nachádza naše SSID tak:

- zastaví sa vyhľadávanie sietí v okolí
- aplikácia sa pripojí k prístupovému bodu pomocou nastavených parametrov
- počká sa kým prístupový bod priradí telefónu IP adresu
- spustí sa sledovanie zmeny stavu Wi-Fi
- zavolá sa obslužná časť aplikácie

##### **Wi-FiStatusChange**

Po zaregistrovaní objektu na sledovanie zmien Wi-Fi stavu, sa pri každej zmene vyvolá metóda *onReceive*. V tejto metóde sa overí či je mobilný telefón stále pripojený k nášmu prístupovému bodu. Ak tak nieje zastaví sa sledovanie zmien stavu Wi-Fi a spustí sa vyhľadávanie sietí v okolí.

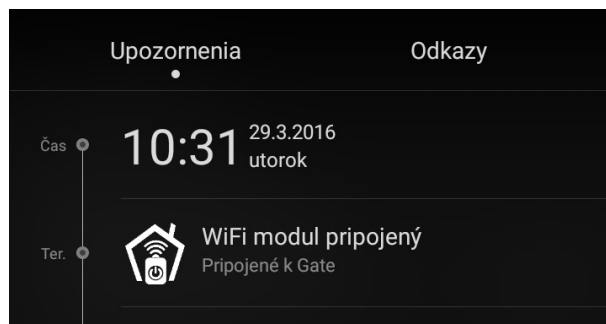
#### • **Notifikácie**

Notifikácia v hornej časti obrazovky slúži ako potvrdenie, že služba beží. Taktiež aj zobrazuje aktuálne dianie na pozadí.

Na nastavenie zobrazenia notifikácií sa používa *NotificationManager*. Notifikácie sa upravujú pomocou zmeny parametrov v *NotificationCompat.Builder*.

Notifikácia zobrazuje, či je mobilný telefón pripojený k prístupovému bodu:

Tu je možné notifikáciu stlačiť a vyvolať obslužnú časť aplikácie. Ak prístupový bod nieje v dosahu mobilného telefónu je zobrazená notifikácia:



Obr. 32: Notifikácia oznamuje, že telefón je pripojený k modulu



Obr. 33: Notifikácia oznamuje vyhľadávanie modulu

Táto notifikácia na stlačenie nereaguje.

### 6.6.3 Obslužná časť

Náplňou obslužnej časti programu je nadviazanie TCP spojenia z Wi-Fi modulom a následná komunikácia. Táto časť je implementovaná v triede *ControllerActivity*, ktorá rozširuje triedu *AppCompatActivity*.

Táto časť je automaticky spustená v prípade, že beží služba na pozadí a mobilný telefón sa pripojí k prístupovému bodu vysielanému Wi-Fi modulom.

#### 6.6.3.1 Grafické rozhranie

Vidíme zobrazené na obr. 35, slúži na interaktívnu komunikáciu užívateľa s modulom. Pozostáva z dvoch častí:

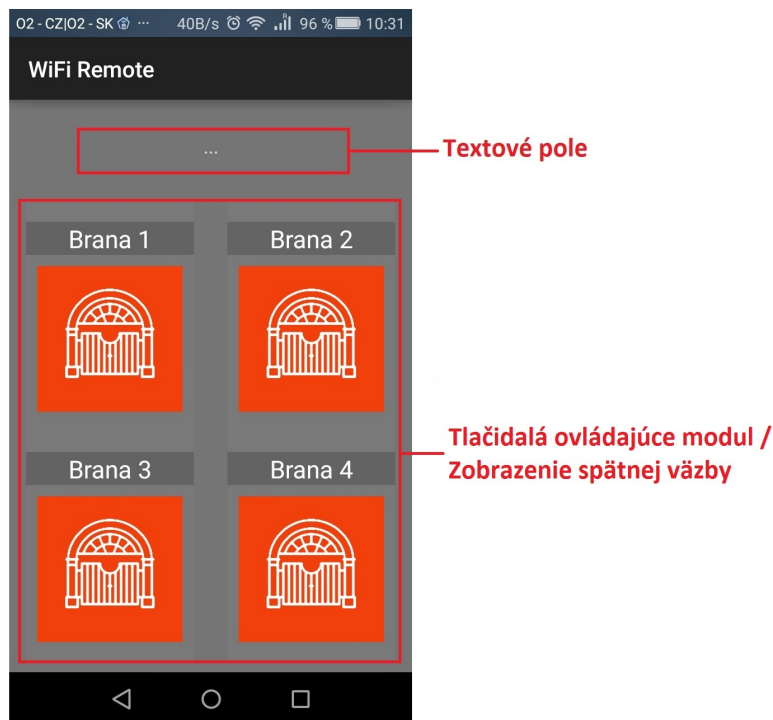
- **Textové pole**

V tejto časti sa zobrazujú textové správy, ktoré prídu zo systému iDúm.

- **Tlačidlá / Spätná väzba**

Tlačidlá slúžia priamo na ovládanie modulu, resp. na spínanie výstupov. Po stlačení tlačidla sa vyšle správa pomocou TCP spojenia do Wi-Fi modulu, kde je preposlaná mikrokon-

troléru, ktorý podľa konfigurácie buď priamo zopne vstup alebo prepošle správu systému iDúm.



Obr. 34: GUI obslužnej časti aplikácie

Spätná väzba modulu je reprezentovaná zmenou ikony tlačidla. Ikona zobrazuje aktuálny priebeh otvárania brány.



Obr. 35: GUI obslužnej časti aplikácie

Pre jednu bránu je spätná väzba reprezentovaná dvomi vstupmi na module.

*Vstup 1* je v log. 1 pokiaľ je brána úplne zatvorená, v momente keď sa brána pootvorí je kontakt rozopnutý a na vstupe je log.0.

Vstup 1	Vstup 2	Stav
0	0	Otvorená
0	1	Otvára / Zatvára sa
1	0	-
1	1	Zatvorená

Tabuľka 2: Definícia spätnej väzby

*Vstup 2* je v stave log. 0 kým je brána plne otvorená, ak sa brána začne zatvárať je kontakt zopnutý a na vstupe je log. 1.

### 6.6.3.2 Pribeh programu

Obslužná časť môže byť vyvolaná dvoma spôsobmi, automaticky po pripojení sa na prístupový bod. V prípade, že sme stále pripojení na prístupový bod a aplikáciu vypneme môžeme ju opäť vyvolať ťuknutím na aktívnu notifikáciu.

- **Inicializácia**

Po vyvolaní tejto aktivity sa v prvom rade inicializujú tlačidlá a textové pole z grafického rozhrania, následne *Wi-FiManager*. V ňom sa overí či je mobilný telefón pripojený na náš prístupový bod. Ak tak je pristúpi sa k vytvoreniu TCP spojenia.

- **Vytvorenie TCP spojenia**

Vytvorenie TCP spojenia je zaistené vo vlastnom vlákne, na tento účel bola vytvorená vnorená trieda *ClientThread*, ktorá rozširuje triedu *Runnable*.

Po spustení tohto vlákna sa vytvorí TCP socket voči Wi-Fi modulu. Na odosielanie informácií do Wi-Fi modulu sa použije *PrintWriter* a na prijímanie správ *BufferedReader*. Nasleduje cyklus, v ktorom sa čaká na prichádzajúce správy. Tento cyklus beží kým je spojenie aktívne.

V aplikácii sú definované dva druhy prichádzajúcich správ:

#### **Spätná väzba**

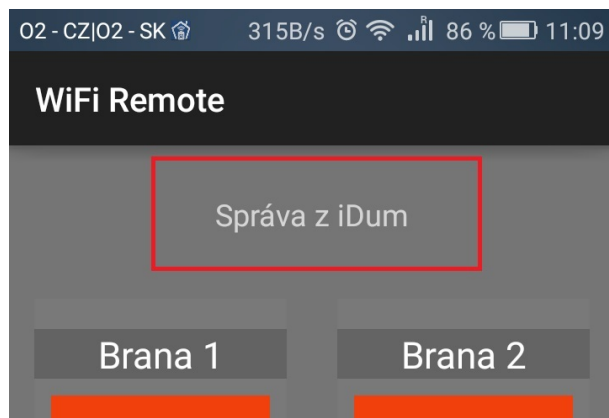
V prípade, že príde správa v ktorej prvý znak je "S" jedná sa o spätnú väzbu z modulu, nasledujúce štyri znaky sú logickou reprezentáciou vstupov na module.

Príklad: S0011 - 00 - brána 2 je otvorená - 11 - brána 1 je zatvorená

Táto správa je spracovaná a podľa nej sa nastaví ikony príslušných tlačidiel aby odpovedali momentálnemu stavu.

#### **Text zo systému iDům**

Ak prvým znakom je "#" znamená to, že sa jedná o textovú správu určenú na zobrazenie v aplikácii, znak "#" je odstránený a zvyšok správy je prepísaný do textového poľa.



Obr. 36: Zobrazená správa zo systému iDum

Ak je stlačené tlačidlo vyšle sa správa pomocou *PrintWriter* a to v nasledujúcej forme:

Prvým znakom je “X” čo indikuje, že sa jedná o stav tlačidiel. Za týmto znakom nasleduje logické vyjadrenie stavu štyroch tlačidiel.

Príklad: X0110 - Tlačidlo 2 a 3 bolo stlačené.

## 7 Popis funkčnosti modulu

V tej to kapitole bude popísané, akým spôsobom sa má zachádzať s modulom a čo všetko sa deje v jednotlivých častiach použitia s názornými príkladmi.

Konfigurácia modulu je určená hodnotami v EEPROM pamäti, ktorú je možné zmeniť pomocou konfigurátora pre systém iDŭm. Tabuľka 3 zobrazuje jednotlivé premenné v pamäti a ich význam v konfigurácii.

Adresa	Popis
0	DevID - Unikátne identifikačné číslo modulu v systéme iDŭm
1	Čas zákmitu tlačidla - hodnota je v ms
2	Čas zopnutia výstupu 1 - hodnota je v t[s]=hodnota/10
3	Čas zopnutia výstupu 2 - hodnota je v t[s]=hodnota/10
4	Čas zopnutia výstupu 3 - hodnota je v t[s]=hodnota/10
5	Čas zopnutia výstupu 4 - hodnota je v t[s]=hodnota/10
6	Povolené tlačidlá - binárna reprezentácia čísla určuje ktoré tlačidlá sú povolené
7	Server ID - ID číslo riadiaceho zariadenia systému iDŭm - nula v prípade použitia bez systému iDŭm

Tabuľka 3: Tabuľka konfigurácie EEPROM pamäte modulu

Každá premenná v pamäti je reprezentovaná jedným bajtom, čo znamená že môže nadobúdať hodnotu 0 - 255. Táto hodnota je zvyčajne reprezentovaná v hexadecimálnom stave (0x00 - 0xFF).

### 7.1 Spustenie

#### 7.1.1 Modul

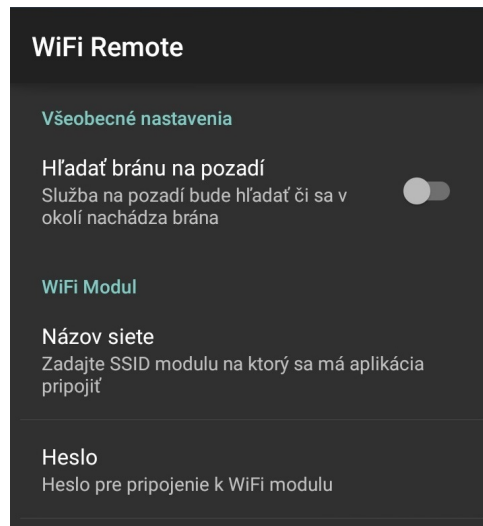
Pred spustením je potrebné pripojiť vstupy a výstupy modulu. V prípade, že modul ma spolupracovať so systémom iDŭm, je potrebné ho pripojiť k spoločnej zbernici iDŭm.

Aby bol modul spustený, je potrebné pripojiť ho na napájacie napätie +5 V. Po pripojení napájania je nutné počkať zhruba 5 s, kým sa inicializuje program na mikrokontroléry ako aj na Wi-Fi module samotnom a začne vysielat v režime prístupového bodu. Po tejto dobe je modul pripravený.

#### 7.1.2 Mobilná aplikácia

Po nainštalovaní mobilnej aplikácie do mobilného telefónu s operačným systémom Android je potrebné aplikáciu spustiť a na prvej obrazovke (obrázok 37), v časti Wi-Fi modul nastaviť názov siete a prihlasovacie heslo k prístupovému bodu modulu.

Pred ďalším krokom je potrebné skontrolovať či je na mobilnom telefóne zapnutá Wi-Fi. Následne môžeme spustiť hľadanie modulu v okolí pomocou tlačidla *Hľadať bránu na pozadí*, v sekcii Všeobecné nastavenia.

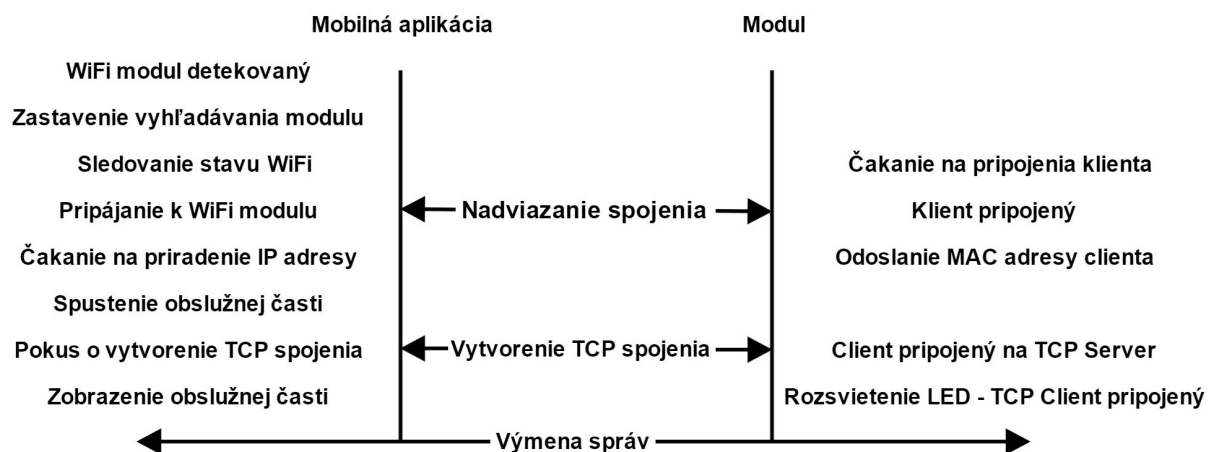


Obr. 37: Nastavenia aplikácie

## 7.2 Ovládanie modulu

Po spustení aplikácie a modulu je všetko pripravené na ovládanie modulu. Predstavme si modelovú situáciu, kde sa nachádzame vo väčšej vzdialenosti od modulu, mimo dosah a následne prichádzame bližšie až do momentu, kedy je prístupový bod aplikáciou rozpoznaný a je k nemu pripojený.

Kým sa nachádzame mimo dosah modulu, na pozadí sa každých 5 sekúnd spustí vyhľadávanie a zoznam dostupných sietí sa aktualizuje, ak sa v ňom nenachádza SSID modulu, tak sa vyhľadávanie opakuje znova. Keď sa mobilný telefón dostane do dosahu modulu prebieha pripájanie. Priebeh pripojenia je zobrazený na obrázku 38.



Obr. 38: Pripojenie k modulu

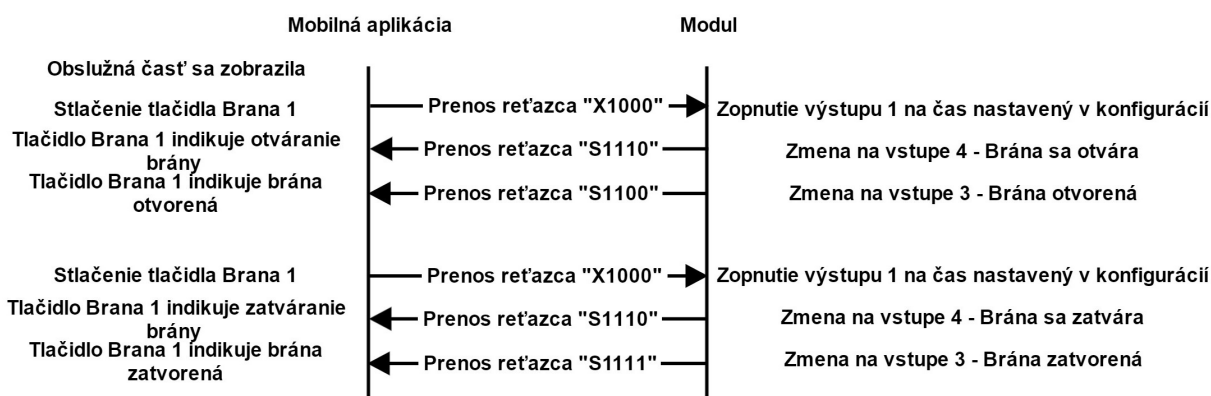
Spojenie medzi mobilnou aplikáciou a modulom je nadviazané a pripravené na komunikáciu.

Na mobilnej aplikácii sa po vytvorení TCP spojenia zobrazí obslužná časť aplikácie, na ktorej je možné modul ovládať. Správanie a spôsob spracovania správ sa následne mierne odlišujú podľa toho či modul je alebo nieje pripojený k systému iDŕm.

### 7.2.1 Modul v režime bez systému iDŕm

Ak chceme modul používať ako samostatnú jednotku bez pripojenia k systému iDŕm. Je potrebné v konfigurácii nastaviť na pozícii 7 - Server ID hodnotu 0. V tomto prípade nebudú správy posielané ďalej do systému iDŕm, ale budú spracované priamo modulom.

Obrázok 39 popisuje modelovú situáciu, kde chceme otvoriť bránu 1 a následne ju opäť zavrieť.



Obr. 39: Prenos správ bez systému iDŕm

Správy vysielané modulom sú vždy posielané všetkým TCP klientom pripojeným na server.

### 7.2.2 Modul pripojený k systému iDŕm

Použitie modulu v spolupráci so systémom iDŕm vyžaduje pripojenie k zbernici a je potrebné nastaviť v konfigurácii na pozícii 7 (Server ID) identifikačné číslo riadiaceho počítača systému iDŕm. Na rozdiel od predchádzajúceho módu, sú správy posielané vždy do hlavného počítača systému iDŕm.

Správy pre systém iDŕm sú definované nasledovne:

Riadiaci počítač systému iDŕm vždy spracuje správu a rozhodne či bude daný požiadavok vykonaný. Keďže má systém k dispozícii MAC adresu klienta, môže na základe tejto informácie rozhodnúť či je daný užívateľ oprávnený určitú akciu vykonať.

Obrázok 40 zobrazuje príklad prenosu správ v prípade, že chceme otvoriť bránu 1.

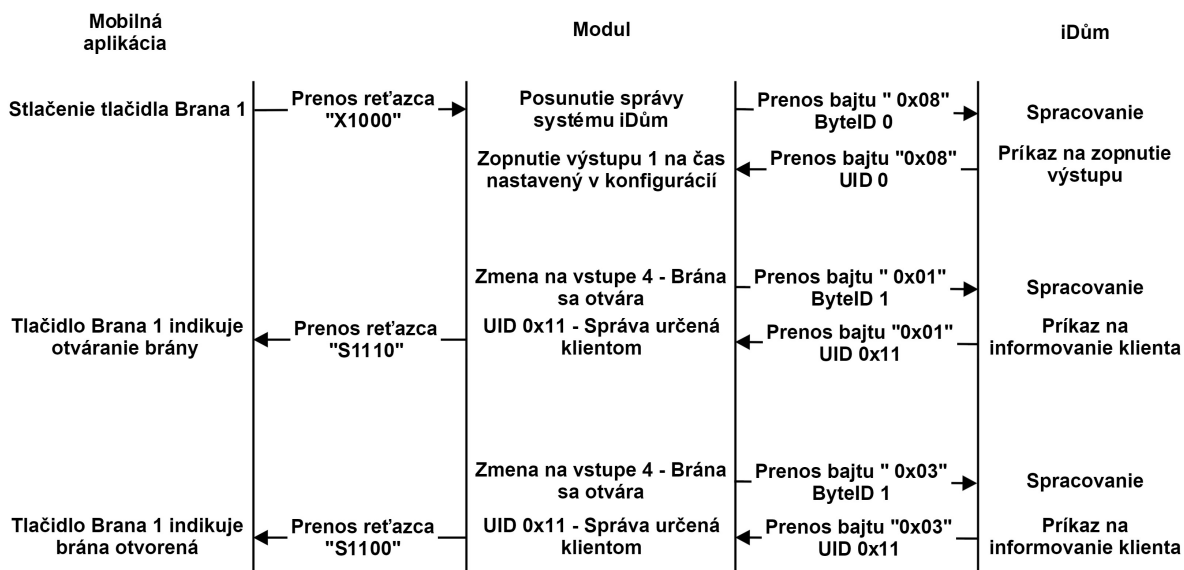
V tomto režime sa na rozdiel od predchádzajúceho využívajú dve špeciálne správy.

Prvou je informovanie systému iDŕm o MAC adrese klienta v momente, keď sa klient sa pripojí a odpojí od modulu. Táto informácia je poslaná do systému iDŕm vo forme poľa znakov.



Správa	Smer	Typ	Byte ID	UID
Stlačenie tlačidla v aplikácii	Modul-iDŭm	Byte	0	0
Zmena na vstupe modulu - spätná väzba	Modul-iDŭm	Byte	1	0
Klient sa pripojil k modulu	Modul-iDŭm	Array	-	0x01
Klient sa odpojil od modulu	Modul-iDŭm	Array	-	0
Informovanie klienta o stave brány	iDŭm-Modul	Byte	0	0x11
Príkaz na otvorenie brány	iDŭm-Modul	Byte	0	0
Textová správa pre klienta	iDŭm-Modul	Array	-	0x11

Tabuľka 4: Správy medzi modulom a systémom iDŭm



Obr. 40: Priebeh otvorenia brány so systémom iDŭm

Druhou správou je možnosť informovať pripojeného klienta o udalosti formou textovej správy, ktorá sa zobrazí na displeji. Táto správa je posiadaná zo systému iDŭm formou poľa znakov. UID tejto správy musí byť nastavené na hodnotu "0x11".

### 7.3 Ukončenie

V prípade, že sa vzdialíme z dosahu prístupového bodu, je spojenie ukončené.

Modul a aplikácia sa vracajú do pôvodného stavu ako je zobrazené na obrázku 41. Mobilná aplikácia opäť vyhľadáva, či sa v okolí nenachádza modul. Modul čaká na pripojenie klienta.

### 7.4 Zabezpečenie

Zabezpečenie voči pripojeniu k modulu a neoprávněnému použitiu je riešené v prvom rade zabezpečením prístupového bodu modulu pomocou WPA2-PSK (AES). Týmto spôsobom je prenos dát medzi mobilným klientom a modulom šifrovaný.



Obr. 41: Vzdialenie sa z dosahu modulu

V prípade, že je modul pripojený k systému iDŕm, je zabezpečenie posilnené posielaním MAC adresy klienta do systému. MAC adresa je odosielaná pri každom pripojení a odpojení od prístupového bodu modulu. V systéme iDŕm môže byť vytvorený zoznam povolených užívateľov, ktorý môžu modul ovládať a sk sa na tomto zozname klient nenachádza nebude jeho požiadavok vykonaný a systém bude upozornený na neznámeho užívateľa.

## 7.5 Android 5+ a prístupové body bez internetového pripojenia

Počas riešenia diplomovej práce sa objavil problém so stabilitou pripojenia k prístupovému bodu modulu, v prípade keď bola aplikácia nainštalovaná na mobilnom telefóne s operačným systémom Android vo verzii 5 a vyššie. Táto nestabilita sa prejavovala samovoľným odpojením od prístupového bodu. V niektorých prípadoch bolo dokonca pripojenie na tento prístupový bod operačným systémom zakázané.

Podľa článku zverejneného na stránke Android Police [13], by malo ísť o zlepšenie užívateľského dojmu pri použití bezdrôtových sietí. Keďže sa predpokladá, že sa užívateľ pripája na Wi-Fi len z dôvodu pripojenia na internet. Pripojenie k prístupovým bodom bez prístupu na internet je považované za zbytočné.

Tento istý problém bol pozorovaný aj pri použití s inými aplikáciami, ako napríklad aplikácia GoPro[14] - ktorá sa pripája na prístupový bod kamery. Tento prístupový bod taktiež neposkytuje pripojenie k internetu.

### **Možné riešenia:**

- Nastavenie statickej IP adresy a DNS serverov Googlu(8.8.8.8 - 8.8.4.4) - pomohlo niektorým užívateľom tento problém vyriešiť [14]
- Použitie mobilného telefónu s OS Android v maximálnej verzii 4.4.4. - pri tejto verzii problém nebol pozorovaný
- Implementácia priamo v aplikácii, ktorá bude sledovať či sa mobilný telefón odpojil od prístupového modulu a v tomto momente spustí opätovné pripájanie.

Posledný bod z týchto riešení bol implementovaný v aplikácii, toto však nieje ideálne riešenie.

## 7.6 Vylepšenie modulu

Modul môže byť v budúcnosti upravený pre zlepšenie celkovej použiteľnosti, bezpečnosti a integrácie z ostatnými časťami systému.

### 7.6.1 Integrácia

Jedným zo smerov, v ktorom by sa mohol modul vylepšiť je v oblasti integrácie nielen so systémom iDŕm, ale aj s domácim prostredím užívateľa.

- Samostatný prístupový bod modulu môže spôsobovať problémy v prípade, že by sa prekrýval s domácimi sieťami užívateľa. To by znamenalo, že by bol aplikáciou nútený byť pripojený k modulu aj v prípade kedy by chcel využívať domáce pripojenie. Tento problém by sa dal riešiť pripojením modulu ako klienta domácej siete.

Aplikácia by sa tak snažila pripojiť k domácej sieti a v nej by rozpoznala prítomnosť modulu. Ak by sa v tejto sieti nachádzal aplikácia by spustila ovládanie brány. V tomto prípade by aplikácia mohla mať aj formu mini-aplikácie, keďže by bol v domácej sieti modul stále prítomný. Týmto spôsobom by mal užívateľ ovládacie prvky vždy po ruke na ploche svojho mobilného telefónu.

Taktiež by sa vyriešil problém s pripojením k prístupovému bodu spomínanému v podkapitole 7.5.

- Ďalším krokom k lepšej integrácii by bol prístup k viacerým častiam systému iDŕm, nie len k modulu a akciám, ktoré sú spojené s jeho ovládaním. To znamená vytvorenie spoločnej jednej aplikácie pre celý systém iDŕm poprípade zahrnutie funkčnosti tejto aplikácie do už existujúcej.

### 7.6.2 Bezpečnosť

Bezpečnosť môže byť v tomto prípade taktiež zlepšená. Momentálne je založená na šifrovanom spojení s prístupovým bodom a informovaní systému iDŕm o MAC adresách klientov, ktorý sú pripojení.

- Bezpečnosť by mohla byť zlepšená odosielaním mena vlastníka telefónu, respektíve mena, používaného so službami Google. Tieto informácie je možné zo systému vytiahnuť a tak okamžite priradiť MAC adresu k menu užívateľa. Keďže užívateľ môže mať viacero mobilných zariadení.
- Zlepšenie práce s MAC adresami, vytvorenie identifikačného čísla užívateľa na základe známych informácií ako je MAC adresa a meno. Následne by sa každá správa týmto číslom označila. Tým by sme docielili väčšiu kontrolu nad aktivitami užívateľa a mohli stanoviť hranice pre jednotlivých užívateľov.

## 8 Záver

Náplňou tejto práce bola realizácia modulu, ktorý by bol súčasťou inteligentného domu iDům, ale mal by byť schopný fungovať aj samostatne. Súčasťou tohto modulu je aj aplikácia pre inteligentné telefóny postavené na platforme Android OS. Hlavnou úlohou tejto aplikácie je detekcia modulu, automatické zobrazenie ovládacej časti a následne samotná komunikácia s modulom.

V teoretickej časti práce sa nachádza zbežný rozbor systému iDům, spôsob akým je navrhnutý a ako sú jednotlivé časti tohto systému zapojené. Taktiež aj spôsob, akým zariadenia medzi sebou komunikujú. Pri spoločnej zbernici tohto systému je popísaný rozbor signalizácie a prenosu správ.

Nasledujúca kapitola už popisuje Wi-Fi modul ESP8266. Tento modul bol využitý na vytvorenie prístupového bodu pomocou, ktorého sa mobilný telefón pripája k modulu. Je tu zobrazený prehľad funkcií, vývojových nástrojov a spôsob programovania.

Kapitola 4 zobrazuje stručný prehľad o mikroprocesoroch PIC ako aj rady PIC18, do ktorej spadá nami použitý mikroprocesor PIC18F46K22. Obsahuje aj popis vývojového prostredia MPLAB X a programátora PICKit 3.

V poslednej teoretickej kapitole je popísaná platforma Android. Obsahuje architektúru operačného systému, náhľad na vývoj aplikácií pre tento systém a popis vývojového prostredia.

Práca pokračuje praktickým riešením zadania. Kde je v prvom rade popísaný návrh modulu a výber vhodných častí. Pokračuje návrhom logického zapojenia, popisom komunikačných protokolov, fyzickým zapojením jednotlivých častí systému a pripojením celého modulu k systému iDům. Druhou časťou tejto kapitoly sú už popísané jednotlivé časti programu vytvoreného pre Wi-Fi modul, mikrokontrolér a mobilnú aplikáciu.

Celý modul bol realizovaný formou zapojenia v nepájivom poli, kde bola týmto spôsobom overená jeho funkčnosť.

Posledná kapitola sa zaoberá modelovými situáciami a spôsobom, akým modul ako celok funguje a ako jednotlivé články komunikujú a spolupracujú.

V závere sú popísané problémy, ktoré sa vyskytli počas vypracovania a smer ďalšieho vývoja.

Modul môže byť v budúcnosti vylepšený v oblasti integrácie so systémom, bezpečnosti a celkového komfortu používania. Návrhy na zlepšenie sú popísané v podkapitole 7.6.

Pri písaní tejto diplomovej práce bola použitá literatúra a zdroje zo zoznamu použitej literatúry. Práca obsahuje aj prílohu vo forme schémy a zoznamu elektronických súčiastok pripojenú na konci práce.

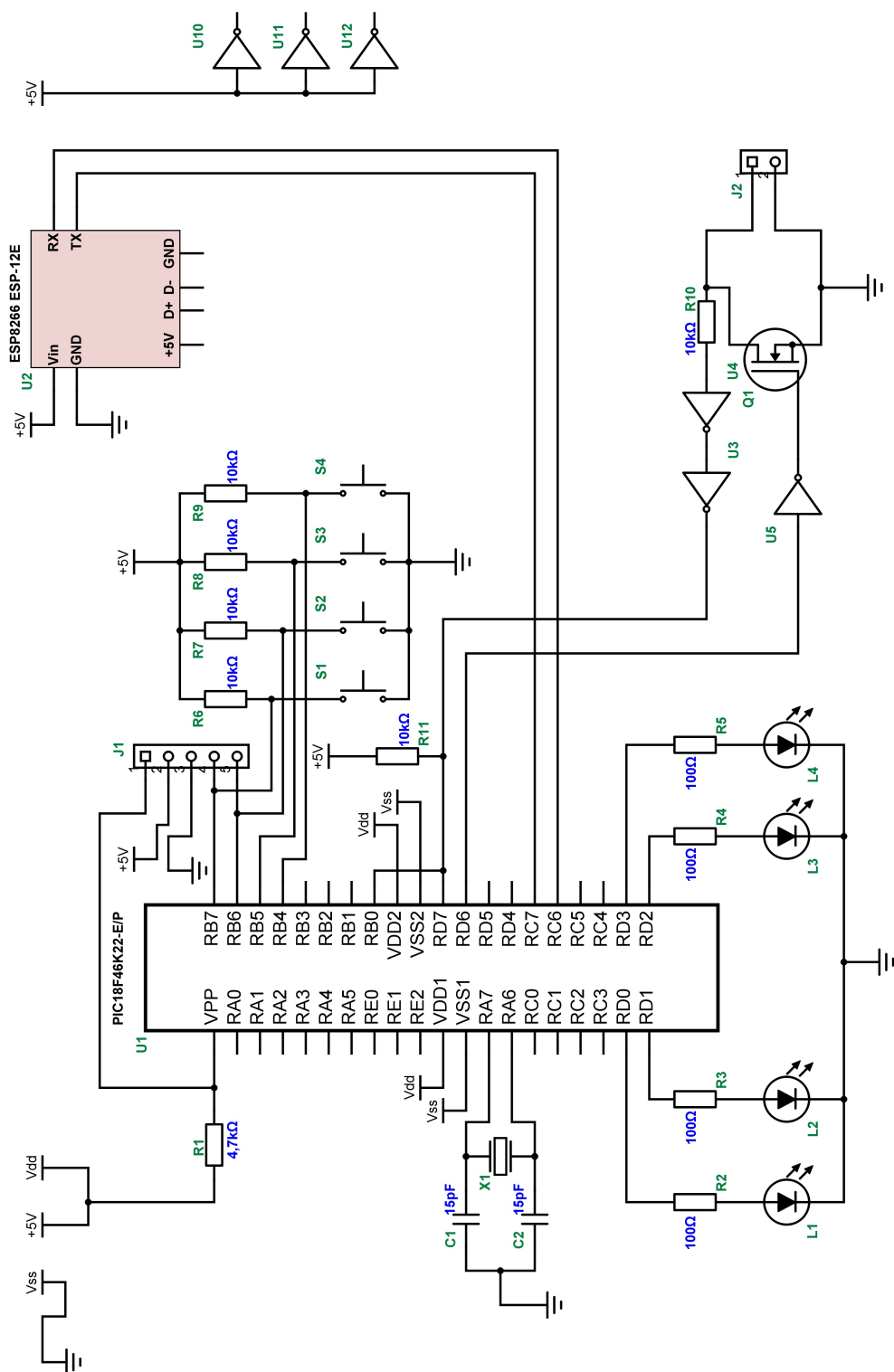
Všetky zdrojové kódy a binárne súbory sú pripojené v elektronickej forme.

## Literatúra

- [1] Jahelka, Michal, *Specifikace iDum*, Ostrava, 23.3.2015.
- [2] ESP8266: *System description.*, 1.4, Espressif Systems Inc., 2016. Dostupné z: <http://espressif.com/products/hardware/esp8266ex/resources>
- [3] NodeMCU: *NodeMCU DEVKIT V1.0* [online]. NodeMCU Team, 2016 [cit. 2016-03-01]. Dostupné z: <https://github.com/nodemcu/nodemcu-devkit-v1.0>
- [4] NodeMCU: *NodeMCU 1.4.0* [online]. NodeMCU Team, 2016 [cit. 2016-03-01]. Dostupné z: <https://github.com/nodemcu/nodemcu-firmware>
- [5] ESP8266 Community Forum: *ESP8266 core for Arduino* [online]. Ivan Grokhotkov, 2016 [cit. 2016-03-01]. Dostupné z: <https://github.com/esp8266/Arduino>
- [6] PICkit<sup>TM</sup> 3 Programmer/Debugger: *User's Guide* [online]. U.S.A: Microchip Technology Incorporated, 2010 [cit. 2016-03-13]. ISBN 978-1-60932-373-8. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/51795B.pdf>
- [7] MPLAB® IDE: MPLAB® X Integrated Development Environment (IDE). *Microchip Technology Inc.* [online]. 2016 [cit. 2016-03-13]. Dostupné z: <http://www.microchip.com/mplab/mplab-x-ide>
- [8] MPLAB® IDE: MPLAB® XC Compilers. *Microchip Technology Inc.* [online]. 2016 [cit. 2016-03-13]. Dostupné z: <http://www.microchip.com/mplab/compilers>
- [9] StatCounter. *Top Mobile Operating Systems Per Country, Apr 2016* [online]. [cit. 2016-04-03]. Dostupné z: [http://gs.statcounter.com/#mobile\\_os-ww-monthly-201601-201601-map](http://gs.statcounter.com/#mobile_os-ww-monthly-201601-201601-map)
- [10] Embedded Linux Wiki. *Android Architecture* [online]. [cit. 2016-04-03]. Dostupné z: [http://elinux.org/Android\\_Architecture](http://elinux.org/Android_Architecture)
- [11] Android Developers: *Application Fundamentals* [online]. [cit. 2016-04-05]. Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- [12] Android Developers: *Android Studio Overview* [online]. [cit. 2016-04-05]. Dostupné z: <http://developer.android.com/tools/studio/index.html>
- [13] Android Police. *Android Now Recognizes Wi-Fi Connections With No Internet Access And Won't Automatically Reconnect To Them* [online]. 2015 [cit. 2016-04-09]. Dostupné z: <http://www.androidpolice.com/2015/03/10/android-5-1-feature-spotlight-android-now-recognizes-wi-fi-connections-with-no-internet-access-and-wont-connect-to-them/>

- [14] DEANMV. *Android Lollipop GoPro App Wi-Fi Issues*. *Reddit: r/GoPro* [online]. 2015 [cit. 2016-04-10]. Dostupné z: [https://www.reddit.com/r/gopro/comments/340ykh/android\\_lollipop\\_gopro\\_app\\_wifi\\_issues](https://www.reddit.com/r/gopro/comments/340ykh/android_lollipop_gopro_app_wifi_issues)

## A Schéma a výpis elektronických súčiastok



Obr. 42: Schéma zapojenia modulu

**Zoznam súčiastok:**

**U1** - mikroprocesor PIC18F46K22

**U2** - Wi-Fi modul ESP8266 ESP-12E

**U3, U4, U5, U10, U11, U12** - Schmittov invertor 74HC14

**J1** - Konektor na pripojenie programátora PICKit 3

**J2** - Konektor na pripojenie k zbernici systému iDům

**Q1** - Unipolárny tranzistor IRLR014

**X1** - Kryštál HC49/US QM 16.000MHZ

**S1, S2, S3, S4** - Tlačidlo TC-12ET-X-A00

**C1, C2** - Keramický kondenzátor CK 15p/50V NPO RM5,08 5

**R1** - 4,7k $\Omega$  rezistor

**R2, R3, R4, R5** - 100 $\Omega$  rezistor

**R6, R7, R8, R9, R10, R11** - 10k $\Omega$  rezistor

**L1, L2, L3, L4** - Svietivá červená LED dioda



## B Elektronická príloha

Súčasťou diplomovej práce je CD, ktoré obsahuje všetky elektronické prílohy.

Adresárova štruktúra:

- Zložka - **Mikrokontroler - PIC18F46K22**

*Zložka - Built*

*Zložka - WiFiModul.X* - Kompletný projekt pre MPLAB X so všetkými zdrojovými súbormi

*Popis.txt* - Stručný popis obsahu zložky a použitých vývojových nástrojov

- Zložka - **Mobilna Aplikacia - AndroidOS**

*Zložka - Andorid Studio Projekt* - Kompletný projekt pre Android Studio IDE so všetkými zdrojovými súbormi

*WiFiRemote.apk* - Skompilovaná aplikácie pre OS Android pripravená na inštaláciu

*Popis.txt* - Stručný popis obsahu zložky a použitých vývojových nástrojov

- Zložka - **WiFi modul - EPS8266**

*TCPServerPIC2.ino* - zdrojový kód / projekt do Arduino IDE

*TCPServerPIC2.ino.nodemcu.bin* - Binárny súbor ktorý môže byť priamo flashnutý na ESP8266 - ESP-12E

*Popis.txt* - Stručný popis obsahu zložky a použitých vývojových nástrojov

- **KRI0080-DiplomovaPraca.pdf** - Diplomová práca v elektronickej forme